PROGRAMMING IN C AND DATA STRUCTURES
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER - I/II

Subject Code	15PCD13/23	IA Marks	20
Number of Lecture Hours/Week	04	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03

CREDITS - 04

Course objectives:

The objectives of this course is to make students to learn basic principles of Problem solving, implementing through C programming language and to design & develop programming skills. To gain knowledge of data structures and their applications.

Module -1 : INTRODUCTION TO C LANGUAGE	**Teaching Hours**
Pseudo code solution to problem, Basic concepts in a C program, Declaration, Assignment & Print statements, Data Types, operators and expressions etc, Programming examples and exercise. **Text 1:** Chapter 2, **and** **Text 2:** 1.1, 1.2, 1.3	**10Hours**
Module -2: BRANCHING AND LOOPING	
Two way selection (if, if-else, nested if-else, cascaded if-else), switch statement, ternary operator? Go to, Loops (For, while-do, do-while) in C, break and continue, Programming examples and exercises. **Text 1:** Chapter 3**.** **&** **Text 2:** 4.4.	**10 Hours**
Module – 3: FUNCTIONS, ARRAYS AND STRINGS	
ARRAYS AND STRINGS: Using an array, Using arrays with Functions, Multi-Dimensional arrays. String: Declaring, Initializing, Printing and reading strings, string manipulation functions, String input and output functions, array of strings, Programming examples and Exercises. **Text 1:** 5.7, & **Text 2**: 7.3, 7.4, chapter 9 **FUNCTIONS**: Functions in C, Argument Passing – call by value, call by reference, Functions and program structure, location of functions, void and parameter less Functions, Recursion, Programming examples and exercises. **Text 1:** 1.7, 1.8, Chapter 4. **Text 2:** 5.1 to 5.4.	**10 Hours**

Module-4: STRUCTURES AND FILE MANAGEMENT	
Basic of structures, structures and Functions, Array of structures, structure Data types, type definition, Defining, opening and closing of files, Input and output operations, Programming examples and exercises. **Text 1:** 6.1 to 6.3. **Text 2**: 10.1 to 10.4, Chapter 11.	**10 Hours**
Module-5: POINTERS AND PREPROCESSORS & Data Structures	
Pointers and address, pointers and functions (call by reference) arguments, pointers and arrays, address arithmetic, character pointer and functions, pointers to pointer ,Initialization of pointer arrays, Dynamic memory allocations methods, Introduction to Preprocessors, compiler control Directives, Programming examples and exercises. **Text 1:** 5.1 to 5.6, 5.8. **Text 2: 12.2,** 12.3**, 13.1 to 13.7**. **Introduction to Data Structures**: Primitive and non primitive data types, Abstract data types, Definition and applications of Stacks, Queues, Linked Lists and Trees. **Text 2 : 14.1, 14.2, 14.11, 14.12, 14.13, 14.15, 14.16, 14.17, 15.1**.	**10 Hours**

Course outcomes: On completion of this course, students are able to

- Achieve Knowledge of design and development of C problem solving skills.
- Understand the basic principles of Programming in C language
- Design and develop modular programming skills.
- Effective utilization of memory using pointer technology
- Understands the basic concepts of pointers and data structures.

Question paper pattern:

- The question paper will have ten questions.
- Each full Question consisting of 16 marks
- There will be **2** full questions(with a **maximum** of **four** sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer **5** full questions, selecting one full question from each module.

Text Books:

1. Brian W. Kernighan and Dennis M. Ritchie: The C Programming Language, 2nd Edition, PHI, 2012.
2. Jacqueline Jones & Keith Harrow: Problem Solving with C, 1st Edition, Pearson 2011.

Reference Books:

1. Vikas Gupta: Computer Concepts and C Programming, Dreamtech Press 2013.
2. R S Bichkar, Programming with C, University Press, 2012.
3. V Rajaraman: Computer Programming in C, PHI, 2013.

COMPUTER PROGRAMMING LABORATORY
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER - I/II

Laboratory Code	**15CPL 16 / 15CPL26**	IA Marks	20
Number of Lecture Hours/Week	01Hr Tutorial (Instructions) + 02 Hours Laboratory	Exam Marks	80
Total Number of Lecture Hours	48	Exam Hours	03

CREDITS - 02

Course objectives: To provide basic principles C programming language. To provide design & develop of C programming skills. To provide practical exposures like designing flowcharts, algorithms, how to debug programs etc**.**

Descriptions (if any):

Demonstration of Personal Computer and its Accessories: **Demonstration and Explanation on Disassembly and Assembly of a Personal Computer by the faculty-in-charge. Students have to prepare a write-up on the same and include it in the Lab record and evaluated.**

Laboratory Session-1: Write-up on Functional block diagram of Computer, CPU, Buses, Mother Board, Chip sets, Operating System & types of OS, Basics of Networking & Topology and NIC.

Laboratory Session-2: Write-up on RAM, SDRAM, FLASH memory, Hard disks, Optical media, CD-ROM/R/RW, DVDs, Flash drives, Keyboard, Mouse, Printers and Plotters. Introduction to flowchart, algorithm and pseudo code.

***Note: These* TWO Laboratory sessions** are used to fill the gap between theory classes and practical sessions. Both sessions are to be evaluated as lab experiments.

Laboratory Experiments:

Implement the following programs with WINDOWS / LINUX platform using appropriate C compiler.

1. Design and develop a flowchart or an algorithm that takes three coefficients (***a***, ***b***, and ***c***) of a Quadratic equation ($ax^2+bx+c=0$) as input and compute all possible roots. Implement a C program for the developed flowchart/algorithm and execute the same to output the possible roots for a given set of coefficients with appropriate messages.

2. Design and develop an algorithm to find the ***reverse*** of an integer number **NUM** and check whether it is PALINDROME or NOT. Implement a C program for the developed algorithm that takes an integer number as input and output the reverse of the same with suitable messages. Ex: Num: **2014**, Reverse: **4102**, Not a Palindrome

3. 3a. Design and develop a flowchart to find the square root of a given number ***N***. Implement a C program for the same and execute for all possible inputs with appropriate messages. Note: **Don't use library function *sqrt(n)*.**

   3b. Design and develop a C program to read a ***year*** as an input and find whether it is ***leap year*** or not. Also consider end of the centuries.

4. Design and develop an algorithm to evaluate polynomial $\mathbf{f(x) = a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0}$, for a given value of ***x*** and its coefficients using Horner's method. Implement a C program for the same and execute the program with different set of values of coefficients and ***x.***

5. Draw the flowchart and Write a C Program to compute **Sin(x)** using Taylor series approximation given by $\mathbf{Sin(x) = x - (x^3/3!) + (x^5/5!) - (x^7/7!) + \ldots\ldots}$
   Compare your result with the built- in Library function. Print both the results with appropriate messages.

6. Develop an algorithm, implement and execute a C program that reads ***N*** integer numbers and arrange them in ascending order using ***Bubble Sort***.

7. Develop, implement and execute a C program that reads two matrices ***A*** (**m x n**) and ***B*** (**p x q**) and Compute product of matrices ***A*** and ***B***. Read matrix ***A*** and matrix ***B*** in row major order and in column major order respectively. Print both the input matrices and resultant matrix with suitable headings and output should be in matrix format only. Program must check the compatibility of orders of the matrices for multiplication. Report appropriate message in case of incompatibility.

8. Develop, implement and execute a C program to search a Name in a list of names using ***Binary searching*** Technique.

9. Write and execute a C program that

i. Implements string copy operation ***STRCOPY***(str1,str2) that copies a string *str1* to another string *str2* without using library function.

ii. Read a *sentence* and print frequency of vowels and total count of consonants.

10.

a. Design and develop a C function ***RightShift****(x ,n)* that takes two integers *x* and *n* as input and returns value of the integer *x* rotated to the right by *n* positions. Assume the integers are unsigned. Write a C program that invokes this function with different values for *x* and *n* and tabulate the results with suitable headings.

b. Design and develop a C function ***isprime***(num) that accepts an integer argument and returns 1 if the argument is prime, a 0 otherwise. Write a C program that invokes this function to generate prime numbers between the given range.

11. Draw the flowchart and write a ***recursive*** **C** function to find the factorial of a number, ***n*!**, defined by *fact*(*n*)=1, if *n*=0. Otherwise *fact*(*n*)=*n***fact*(*n*-1). Using this function, write a C program to compute the binomial coefficient ${}_nC_r$. Tabulate the results for different values of ***n*** and ***r*** with suitable messages.

12. Given two university information files "**studentname.txt**" and "**usn.txt**" that contains students Name and USN respectively. Write a C program to create a new file called "**output.txt**" and copy the content of files "studentname.txt" and "usn.txt" into output file in the sequence shown below . Display the contents of output file "output.txt" on to the screen.

Student Name	USN
Name 1	USN1
Name 2	USN2
….	….
….	….

Heading

13. Write a C program to maintain a record of **n** student details using an array of structures with four fields (Roll number, Name, Marks, and Grade). Assume appropriate data type for each field. Print the marks of the student, given the student name as input.

14. Write a C program using pointers to compute the sum, mean and standard deviation of all elements stored in an array of **n** real numbers.

Course outcomes:

- **Gaining Knowledge on various parts of a computer.**
- **Able to draw flowcharts and write algorithms**
- **Able design and development of C problem solving skills.**
- **Able design and develop modular programming skills.**
- **Able to trace and debug a program**

Conduction of Practical Examination:

1. All laboratory experiments (nos) are to be included for practical examination.
2. Students are allowed to pick one experiment from the lot.
3. Strictly follow the instructions as printed on the cover page of answer script for breakup of marks
4. **Change of experiment is allowed only once and 15% Marks allotted to the procedure part to be made zero.**

Analog and Digital Electronics

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER - III

Subject Code	**15CS32**	**IA Marks**	**20**
Number of Lecture Hours/Week	**04**	**Exam Marks**	**80**
Total Number of Lecture Hours	**50**	**Exam Hours**	**03**

CREDITS – 04

Course objectives: This course will enable students to

- Recall and Recognize construction and characteristics of JFETs and MOSFETs and differentiate with BJT
- Demonstrate and Analyze Operational Amplifier circuits and their applications
- Describe, Illustrate and Analyze Combinational Logic circuits, Simplification of Algebraic Equations using Karnaugh Maps and Quine McClusky Techniques.
- Describe and Design Decoders, Encoders, Digital multiplexers, Adders and Subtractors, Binary comparators, Latches and Master-Slave Flip-Flops.
- Describe, Design and Analyze Synchronous and Asynchronous Sequential
- Explain and design registers and Counters, A/D and D/A converters.

Module -1	**Teaching Hours**
Field Effect Transistors: Junction Field Effect Transistors, MOSFETs, Differences between JFETs and MOSFETs, Biasing MOSFETs, FET Applications, CMOS Devices. Wave-Shaping Circuits: Integrated Circuit(IC) Multivibrators. **Introduction to Operational Amplifier**: Ideal v/s practical Opamp, Performance Parameters, **Operational Amplifier Application Circuits**:Peak Detector Circuit, Comparator, Active Filters, Non-Linear Amplifier, Relaxation Oscillator, Current-To-Voltage Converter, Voltage-To-Current Converter. **Text book 1:- Ch 5: 5.2, 5.3, 5.5, 5.8, 5.9, 5.1.Ch13: 13.10.Ch 16: 16.3, 16.4. Ch 17: 7.12, 17.14, 17.15, 17.18, 17.19, 17.20, 17.21.**	**10 Hours**
Module -2	
The Basic Gates: Review of Basic Logic gates, Positive and Negative Logic, Introduction to HDL. **Combinational Logic Circuits:** Sum-of-Products Method, Truth Table to Karnaugh Map, Pairs Quads, and Octets, Karnaugh Simplifications, Don't-care Conditions, Product-of-sums Method, Product-of-sums simplifications, Simplification by Quine-McClusky Method, Hazards and Hazard covers, HDL Implementation Models. **Text book 2:- Ch 2: 2.4, 2.5. Ch3: 3.2 to 3.11.**	**10 Hours**
Module – 3	

Data-Processing Circuits: Multiplexers, Demultiplexers, 1-of-16 Decoder, BCD to Decimal Decoders, Seven Segment Decoders, Encoders, Exclusive-OR Gates, Parity Generators and Checkers, Magnitude Comparator, Programmable Array Logic, Programmable Logic Arrays, HDL Implementation of Data Processing Circuits. Arithmetic Building Blocks, Arithmetic Logic Unit **Flip- Flops:** RS Flip-Flops, Gated Flip-Flops, Edge-triggered RS FLIP-FLOP, Edge-triggered D FLIP-FLOPs, Edge-triggered JK FLIP-FLOPs. **Text book 2:- Ch 4:- 4.1 to 4.9, 4.11, 4.12, 4.14.Ch 6:-6.7, 6.10.Ch 8:- 8.1 to 8.5.**	**10 Hours**
Module-4	
Flip- Flops: FLIP-FLOP Timing, JK Master-slave FLIP-FLOP, Switch Contact Bounce Circuits, Various Representation of FLIP-FLOPs, HDL Implementation of FLIP-FLOP. **Registers:** Types of Registers, Serial In - Serial Out, Serial In - Parallel out, Parallel In - Serial Out, Parallel In - Parallel Out, Universal Shift Register, Applications of Shift Registers, Register implementation in HDL. **Counters:** Asynchronous Counters, Decoding Gates, Synchronous Counters, Changing the Counter Modulus. (**Text book 2:- Ch 8: 8.6, 8.8, 8.9, 8.10, 8.13. Ch 9: 9.1 to 9.8. Ch 10: 10.1 to 10.4**	**10 Hours**
Module-5	
Counters: Decade Counters, Presettable Counters, Counter Design as a Synthesis problem, A Digital Clock, Counter Design using HDL. **D/A Conversion and A/D Conversion:** Variable, Resistor Networks, Binary Ladders, D/A Converters, D/A Accuracy and Resolution, A/D Converter-Simultaneous Conversion, A/D Converter-Counter Method, Continuous A/D Conversion, A/D Techniques, Dual-slope A/D Conversion, A/D Accuracy and Resolution. **Text book 2:- Ch 10: 10.5 to 10.9. Ch 12: 12.1 to 12.10.**	**10 Hours**
Course outcomes:	
After Studying this course, students will be able to • Acquire knowledge of o JFETs and MOSFETs , Operational Amplifier circuits and their applications. o Combinational Logic, Simplification Techniques using Karnaugh Maps, Quine McClusky technique. o Operation of Decoders, Encoders, Multiplexers, Adders and Subtractors. o Working of Latches, Flip-Flops, Designing Registers, Counters, A/D and D/A Converters. • Analyze the performance of o JFETs and MOSFETs , Operational Amplifier circuits o Simplification Techniques using Karnaugh Maps, Quine McClusky Technique. o Synchronous and Asynchronous Sequential Circuits. Apply the knowledge gained in the design of Counters, Registers and A/D & D/A converters	
Graduate Attributes (as per NBA) 1. Engineering Knowledge 2. Design/Development of Solutions(partly) 3. Modern Tool Usage 4. Problem Analysis	

Question paper pattern:

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Anil K Maini, Varsha Agarwal: Electronic Devices and Circuits, Wiley, 2012.
2. Donald P Leach, Albert Paul Malvino & Goutam Saha: Digital Principles and Applications, 8th Edition, Tata McGraw Hill, 2015

Reference Books:

1. Stephen Brown, Zvonko Vranesic: Fundamentals of Digital Logic Design with VHDL, 2nd Edition, Tata McGraw Hill, 2005.
2. R D Sudhaker Samuel: Illustrative Approach to Logic Design, Sanguine-Pearson, 2010.
3. M Morris Mano: Digital Logic and Computer Design, 10th Edition, Pearson, 2008.

DATA STRUCTURES AND APPLICATIONS

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER - III

Subject Code	**15CS33**	**IA Marks**	**20**
Number of Lecture Hours/Week	**04**	**Exam Marks**	**80**
Total Number of Lecture Hours	**50**	**Exam Hours**	**03**

CREDITS - 04

Course objectives: This course will enable students to

- Explain fundamentals of data structures and their applications essential for programming/problem solving
- Analyze Linear Data Structures: Stack, Queues, Lists
- Analyze Non-Linear Data Structures: Trees, Graphs
- Analyze and Evaluate the sorting & searching algorithms
- Assess appropriate data structure during program development/Problem Solving

Module -1	**Teaching Hours**
Introduction: Data Structures, Classifications (Primitive & Non Primitive), Data structure Operations, Review of Arrays, Structures, Self-Referential Structures, and Unions. Pointers and Dynamic Memory Allocation Functions. Representation of Linear Arrays in Memory, Dynamically allocated arrays, **Array Operations**: Traversing, inserting, deleting, searching, and sorting. Multidimensional Arrays, Polynomials and Sparse Matrices. **Strings:** Basic Terminology, Storing, Operations and Pattern Matching algorithms. Programming Examples. **Text 1: Ch 1: 1.2, Ch 2: 2.2 -2.7** **Text 2: Ch 1: 1.1 -1.4, Ch 3: 3.1-3.3,3.5,3.7, Ch 4: 4.1-4.9,4.14** **Ref 3: Ch 1: 1.4**	**10 Hours**
Module -2	
Stacks and Queues ***Stacks:*** Definition, Stack Operations, Array Representation of Stacks, Stacks using Dynamic Arrays, Stack Applications: Polish notation, Infix to postfix conversion, evaluation of postfix expression, **Recursion** - Factorial, GCD, Fibonacci Sequence, Tower of Hanoi, Ackerman's function. **Queues:** Definition, Array Representation, Queue Operations, Circular Queues, Circular queues using Dynamic arrays, Dequeues, Priority Queues, A Mazing Problem. Multiple Stacks and Queues. Programming Examples. **Text 1: Ch 3: 3.1 -3.7** **Text 2: Ch 6: 6.1 -6.3, 6.5, 6.7-6.10, 6.12, 6.13**	**10 Hours**
Module - 3	

Linked Lists: Definition, Representation of linked lists in Memory, Memory allocation; Garbage Collection. Linked list operations: Traversing, Searching, Insertion, and Deletion. Doubly Linked lists, Circular linked lists, and header linked lists. Linked Stacks and Queues. Applications of Linked lists – Polynomials, Sparse matrix representation. Programming Examples **Text 1: Ch 4: 4.1 -4.8 except 4.6** **Text 2: Ch 5: 5.1 – 5.10**	**10 Hours**
Module-4	
Trees: Terminology, Binary Trees, Properties of Binary trees, Array and linked Representation of Binary Trees, Binary Tree Traversals - Inorder, postorder, preorder; Additional Binary tree operations. Threaded binary trees, Binary Search Trees – Definition, Insertion, Deletion, Traversal, Searching, Application of Trees-Evaluation of Expression, Programming Examples **Text 1: Ch 5: 5.1 –5.5, 5.7** **Text 2: Ch 7: 7.1 – 7.9**	**10 Hours**
Module-5	
Graphs: Definitions, Terminologies, Matrix and Adjacency List Representation Of Graphs, Elementary Graph operations, Traversal methods: Breadth First Search and Depth First Search. **Sorting and Searching**: Insertion Sort, Radix sort, Address Calculation Sort. **Hashing:** Hash Table organizations, Hashing Functions, Static and Dynamic Hashing. **Files and Their Organization:** Data Hierarchy, File Attributes, Text Files and Binary Files, Basic File Operations, File Organizations and Indexing **Text 1: Ch 6: 6.1 –6.2, Ch 7:7.2, Ch 8:8.1-8.3** **Text 2: Ch 8: 8.1 – 8.7, Ch 9:9.1-9.3,9.7,9.9** **Reference 2: Ch 16: 16.1 - 16.7**	**10 Hours**
Course outcomes:	
After studying this course, students will be able to: • Acquire knowledge of - Various types of data structures, operations and algorithms. - Sorting and searching operations. - File structures. • Analyse the performance of - Stack, Queue, Lists, Trees, Graphs, Searching and Sorting techniques. • Implement all the applications of Data structures in a high-level language. • Design and apply appropriate data structures for solving computing problems.	
Graduate Attributes (as per NBA) 1. Engineering Knowledge 2. Design/Development of Solutions 3. Conduct Investigations of Complex Problems 4. Problem Analysis	

Question paper pattern:

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Fundamentals of Data Structures in C - Ellis Horowitz and Sartaj Sahni, 2nd edition, Universities Press,2014
2. Data Structures - Seymour Lipschutz, Schaum's Outlines, Revised 1st edition, McGraw Hill, 2014

Reference Books:

1. Data Structures: A Pseudo-code approach with C –Gilberg & Forouzan, 2nd edition, Cengage Learning, 2014.
2. Data Structures using C, , Reema Thareja, 3rd edition Oxford press, 2012.
3. An Introduction to Data Structures with Applications- Jean-Paul Tremblay & Paul G. Sorenson, 2nd Edition, McGraw Hill, 2013.
4. Data Structures using C - A M Tenenbaum, PHI, 1989.
5. Data Structures and Program Design in C - Robert Kruse, 2nd edition, PHI, 1996.

COMPUTER ORGANIZATION

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER - III

Subject Code	**15CS34**	**IA Marks**	**20**
Number of Lecture Hours/Week	**04**	**Exam Marks**	**80**
Total Number of Lecture Hours	**50**	**Exam Hours**	**03**

CREDITS – 04

Course objectives:

This course will enable students to

- Understand the basics of computer organization: structure and operation of computers and their peripherals.
- Understand the concepts of programs as sequences or machine instructions.
- Expose different ways of communicating with I/O devices and standard I/O interfaces.
- Describe hierarchical memory systems including cache memories and virtual memory.
- Describe arithmetic and logical operations with integer and floating-point operands.
- Understand basic processing unit and organization of simple processor, concept of pipelining and other large computing systems.

Module -1	**Teaching Hours**
Basic Structure of Computers: Basic Operational Concepts, Bus Structures, Performance – Processor Clock, Basic Performance Equation, Clock Rate, Performance Measurement. **Machine Instructions and Programs:** Memory Location and Addresses, Memory Operations, Instructions and Instruction Sequencing, Addressing Modes, Assembly Language, Basic Input and Output Operations, Stacks and Queues, Subroutines, Additional Instructions, Encoding of Machine Instructions **Textbook 1: Ch 1: 1.3, 1.4, 1.6.1, 1.6.2, 1.6.4, 1.6.7. Ch 2: 2.2 to 2.10, 2.12**	**10Hours**
Module -2	
Input/Output Organization: Accessing I/O Devices, Interrupts – Interrupt Hardware, Enabling and Disabling Interrupts, Handling Multiple Devices, Controlling Device Requests, Exceptions, Direct Memory Access, Buses, Interface Circuits, Standard I/O Interfaces – PCI Bus, SCSI Bus, USB. **Textbook 1: Ch 4: 4.1, 4.2: 4.2.1 to 4.2.5, 4.4 to 4.7.**	**10 Hours**
Module – 3	
Memory System: Basic Concepts, Semiconductor RAM Memories, Read Only Memories, Speed, Size, and Cost, Cache Memories – Mapping Functions, Replacement Algorithms, Performance Considerations, Virtual Memories, Secondary Storage. **Textbook 1: Ch 5: 5.1 to 5.4, 5.5.1, 5.5.2, 5.6, 5.7, 5.9**	**10 Hours**
Module-4	

Arithmetic: Numbers, Arithmetic Operations and Characters, Addition and Subtraction of Signed Numbers, Design of Fast Adders, Multiplication of Positive Numbers, Signed Operand Multiplication, Fast Multiplication, Integer Division, Floating-point Numbers and Operations. **Textbook 1: Ch 2: 2.1, Ch 6: 6.1 to 6.7**	**10 Hours**
Module-5	
Basic Processing Unit: Some Fundamental Concepts, Execution of a Complete Instruction, Multiple Bus Organization, Hard-wired Control, Micro programmed Control. **Embedded Systems and Large Computer Systems:** Examples of Embedded Systems, Processor chips for embedded applications, Simple Microcontroller. **The structure of General-Purpose Multiprocessors**. **Textbook 1: Ch 7: 7.1 to 7.5, Ch 9:9.1 to 9.3, Ch 12:12.3**	**10 Hours**

Course outcomes:

After studying this course, students will be able to:

- Acquire knowledge of
  - The basic structure of computers & machine instructions and programs, Addressing Modes, Assembly Language, Stacks, Queues and Subroutines.
  - Input/output Organization such as accessing I/O Devices, Interrupts.
  - Memory system basic Concepts, Semiconductor RAM Memories, Static memories, Asynchronous DRAMS, Read Only Memories, Cache Memories and Virtual Memories.
  - Some Fundamental Concepts of Basic Processing Unit, Execution of a Complete Instruction, Multiple Bus Organization, Hardwired Control and Micro programmed Control.
  - Pipelining, embedded and large computing system architecture.
- Analyse and design arithmetic and logical units.
- Apply the knowledge gained in the design of Computer.
- Design and evaluate performance of memory systems
- Understand the importance of life-long learning

Graduate Attributes (as per NBA)

1. Engineering Knowledge
2. Problem Analysis
3. Life-Long Learning

Question paper pattern:

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Carl Hamacher, Zvonko Vranesic, Safwat Zaky: Computer Organization, 5th Edition, Tata McGraw Hill, 2002.

Reference Books:

1. William Stallings: Computer Organization & Architecture, 9th Edition, Pearson, 2015.

UNIX AND SHELL PROGRAMMING

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER – III

Subject Code	**15CS35**	**IA Marks**	**20**
Number of Lecture Hours/Week	**04**	**Exam Marks**	**80**
Total Number of Lecture Hours	**50**	**Exam Hours**	**03**

CREDITS – 04

Course objectives: This course will enable students to

- Understand the UNIX Architecture, File systems and use of basic Commands.
- Use of editors and Networking commands.
- Understand Shell Programming and to write shell scripts.
- Understand and analyze UNIX System calls, Process Creation, Control & Relationship.

Module -1	**Teaching Hours**
Introduction, Brief history. Unix Components/Architecture. Features of Unix. The UNIX Environment and UNIX Structure, Posix and Single Unix specification. The login prompt. General features of Unix commands/ command structure. Command arguments and options. Understanding of some basic commands such as echo, printf, ls, who, date, passwd, cal, Combining commands. Meaning of Internal and external commands. The type command: knowing the type of a command and locating it. The man command knowing more about Unix commands and using Unix online manual pages. The man with keyword option and whatis. The more command and using it with other commands. Knowing the user terminal, displaying its characteristics and setting characteristics. Managing the non-uniform behaviour of terminals and keyboards. The root login. Becoming the super user: su command. The /etc/passwd and /etc/shadow files. Commands to add, modify and delete users. **Topics from chapter 2 , 3 and 15 of text book 1,chapter 1 from text book 2**	**10Hours**
Module -2	
Unix files. Naming files. Basic file types/categories. Organization of files. Hidden files. Standard directories. Parent child relationship. The home directory and the HOME variable. Reaching required files- the PATH variable, manipulating the PATH, Relative and absolute pathnames. Directory commands – pwd, cd, mkdir, rmdir commands. The dot (.) and double dots (..) notations to represent present and parent directories and their usage in relative path names. File related commands – cat, mv, rm, cp, wc and od commands. File attributes and permissions and knowing them. The ls command with options. Changing file permissions: the relative and absolute permissions changing methods. Recursively changing file permissions. Directory permissions. **Topics from chapters 4, 5 and 6 of text book 1**	**10Hours**

Module – 3	
The vi editor. Basics. The .exrc file. Different ways of invoking and quitting vi. Different modes of vi. Input mode commands. Command mode commands. The ex mode commands. Illustrative examples Navigation commands. Repeat command. Pattern searching. The search and replace command. The set, map and abbr commands. Simple examples using these commands. The shells interpretive cycle. Wild cards and file name generation. Removing the special meanings of wild cards. Three standard files and redirection. Connecting commands: Pipe. Splitting the output: tee. Command substitution. Basic and Extended regular expressions. The grep, egrep. Typical examples involving different regular expressions. **Topics from chapters 7, 8 and 13 of text book 1. Topics from chapter 2 and 9 ,10 of text book 2**	**10Hours**
Module-4	
Shell programming. Ordinary and environment variables. The .profile. Read and readonly commands. Command line arguments. exit and exit status of a command. Logical operators for conditional execution. The test command and its shortcut. The if, while, for and case control statements. The set and shift commands and handling positional parameters. The here (<<) document and trap command. Simple shell program examples. File inodes and the inode structure. File links – hard and soft links. Filters. Head and tail commands. Cut and paste commands. The sort command and its usage with different options. The umask and default file permissions. Two special files /dev/null and /dev/tty. **Topics from chapter 11, 12, 14 of text book 1,chapter 17 from text book2**	**10Hours**
Module-5	
Meaning of a process. Mechanism of process creation. Parent and child process. The ps command with its options. Executing a command at a specified point of time: at command. Executing a command periodically: cron command and the crontab file.. Signals. The nice and nohup commands. Background processes. The bg and fg command. The kill command. The find command with illustrative example. Structure of a perl script. Running a perl script. Variables and operators. String handling functions. Default variables - $_ and $. – representing the current line and current line number. The range operator. Chop() and chomp() functions. Lists and arrays. The @-variable. The splice operator, push(), pop(), split() and join(). File handles and handling file – using open(), close() and die () functions.. Associative arrays – keys and value functions. Overview of decision making loop control structures – the foreach. Regular expressions – simple and multiple search patterns. The match and substitute operators. Defining and using subroutines. **Topics from chapter 9 and 19 of text book 1. Topics from chapter 11 of reference book 1**	**10Hours**

Course outcomes:

After studying this course, students will be able to:

- Explain multi user OS UNIX and its basic features
- Interpret UNIX Commands, Shell basics, and shell environments
- Design and develop shell programming, communication, System calls and terminology.
- Design and develop UNIX File I/O and UNIX Processes.
- Perl script writing

Graduate Attributes (as per NBA)

1. Engineering Knowledge
2. Environment and Sustainability
3. Design/Development of Solutions

Question paper pattern:

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Sumitabha Das., Unix Concepts and Applications., 4th Edition., Tata McGraw Hill
2. Behrouz A. Forouzan, Richard F. Gilberg : UNIX and Shell Programming- Cengage Learning – India Edition. 2009.

Reference Books:

1. M.G. Venkatesh Murthy: UNIX & Shell Programming**,** Pearson Education.
2. Richard Blum , Christine Bresnahan : Linux Command Line and Shell Scripting Bible, 2nd Edition , Wiley,2014.

DISCRETE MATHEMATICAL STRUCTURES

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER – III

Subject Code	**15CS36**	**IA Marks**	**20**
Number of Lecture Hours/Week	**04**	**Exam Marks**	**80**
Total Number of Lecture Hours	**50**	**Exam Hours**	**03**

CREDITS – 04

Course objectives: This course will enable students to

- Prepare for a background in abstraction, notation, and critical thinking for the mathematics most directly related to computer science.
- Understand and apply logic, relations, functions, basic set theory, countability and counting arguments, proof techniques,
- Understand and apply mathematical induction, combinatorics, discrete probability, recursion, sequence and recurrence, elementary number theory
- Understand and apply graph theory and mathematical proof techniques.

Module -1	**Teaching Hours**
Fundamentals of Logic: Basic Connectives and Truth Tables, Logic Equivalence – The Laws of Logic, Logical Implication – Rules of Inference. The Use of Quantifiers, Quantifiers, Definitions and the Proofs of Theorems, **Textbook 1: Ch 2**	**10Hours**
Module -2	
Properties of the Integers: Mathematical Induction, The Well Ordering Principle – Mathematical Induction, Recursive Definitions. **Fundamental Principles of Counting:** The Rules of Sum and Product, Permutations, Combinations – The Binomial Theorem, Combinations with Repetition, **Textbook 1: Ch 4: 4.1, 4.2 Ch 1.**	**10 Hours**
Module – 3	
Relations and Functions: Cartesian Products and Relations, Functions – Plain and One-to-One, Onto Functions. The Pigeon-hole Principle, Function Composition and Inverse Functions. **Properties of Relations**, Computer Recognition – Zero-One Matrices and Directed Graphs, Partial Orders – Hasse Diagrams, Equivalence Relations and Partitions. **Textbook 1: Ch 5:5.1 to 5.3, 5.5, 5.6, Ch 7:7.1 to 7.4**	**10 Hours**
Module-4	

The Principle of Inclusion and Exclusion: The Principle of Inclusion and Exclusion, Generalizations of the Principle, Derangements – Nothing is in its Right Place, Rook Polynomials. **Recurrence Relations:** First Order Linear Recurrence Relation, The Second Order Linear Homogeneous Recurrence Relation with Constant Coefficients. **Textbook 1: Ch 8: 8.1 to 8.4, Ch 10:10.1 to 10.2**	**10 Hours**
Module-5	
Introduction to Graph Theory: Definitions and Examples, Sub graphs, Complements, and Graph Isomorphism, Vertex Degree, Euler Trails and Circuits , **Trees**: Definitions, Properties, and Examples, Routed Trees, Trees and Sorting, Weighted Trees and Prefix Codes **Textbook 1: Ch 11: 11.1 to 11.3, Ch 12: 12.1 to 12.4**	**10 Hours**

Course outcomes:

After studying this course, students will be able to:

1. Verify the correctness of an argument using propositional and predicate logic and truth tables.
2. Demonstrate the ability to solve problems using counting techniques and combinatorics in the context of discrete probability.
3. Solve problems involving recurrence relations and generating functions.
4. Construct proofs using direct proof, proof by contraposition, proof by contradiction, proof by cases, and mathematical induction.
5. Explain and differentiate graphs and trees

Graduate Attributes (as per NBA)

1. Engineering Knowledge
2. Problem Analysis
3. Conduct Investigations of Complex Problems

Question paper pattern:

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Ralph P. Grimaldi: Discrete and Combinatorial Mathematics, , 5th Edition, Pearson Education. 2004.

Reference Books:

1. Basavaraj S Anami and Venakanna S Madalli: Discrete Mathematics – A Concept based approach, Universities Press, 2016
2. Kenneth H. Rosen: Discrete Mathematics and its Applications, 6th Edition, McGraw Hill, 2007.
3. Jayant Ganguly: A Treatise on Discrete Mathematical Structures, Sanguine-Pearson, 2010.
4. D.S. Malik and M.K. Sen: Discrete Mathematical Structures: Theory and Applications, Thomson, 2004.
5. Thomas Koshy: Discrete Mathematics with Applications, Elsevier, 2005, Reprint 2008.

ANALOG AND DIGITAL ELECTRONICS LABORATORY
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER - III

Laboratory Code	**15CSL37**	**IA Marks**	**20**
Number of Lecture Hours/Week	**01I + 02P**	**Exam Marks**	**80**
Total Number of Lecture Hours	**40**	**Exam Hours**	**03**

CREDITS – 02

Course objectives: This laboratory course enable students to get practical experience in design, assembly and evaluation/testing of

- Analog components and circuits including Operational Amplifier, Timer, etc.
- Combinational logic circuits.
- Flip - Flops and their operations
- Counters and Registers using Flip-flops.
- Synchronous and Asynchronous Sequential Circuits.
- A/D and D/A Converters

Descriptions (if any)

Any simulation package like MultiSim / P-spice /Equivalent software may be used.
Faculty-in-charge should demonstrate and explain the required hardware components and their functional Block diagrams, timing diagrams etc. Students have to prepare a write-up on the same and include it in the Lab record and to be evaluated.

Laboratory Session-1: Write-upon analog components; functional block diagram, Pin diagram (if any), waveforms and description. The same information is also taught in theory class; this helps the students to understand better.

Laboratory Session-2: Write-upon Logic design components, pin diagram (if any), Timing diagrams, etc. The same information is also taught in theory class; this helps the students to understand better.

Note: These **TWO Laboratory sessions** are used to fill the gap between theory classes and practical sessions. Both sessions are to be evaluated for 20 marks as lab experiments.

Laboratory Experiments:

1. a) Design and construct a Schmitt trigger using Op-Amp for given UTP and LTP values and demonstrate its working.
   b) Design and implement a Schmitt trigger using Op-Amp using a simulation package for two sets of UTP and LTP values and demonstrate its working.

2. a) Design and construct a rectangular waveform generator (Op-Amp relaxation oscillator) for given frequency and demonstrate its working.
   b) Design and implement a rectangular waveform generator (Op-Amp relaxation oscillator) using a simulation package and demonstrate the change in frequency when all resistor values are doubled.

3. Design and implement an Astable multivibrator circuit using 555 timer for a given frequency and duty cycle.

NOTE: hardware and software results need to be compared

Continued:

4. Design and implement Half adder, Full Adder, Half Subtractor, Full Subtractor using basic gates.

5. a) Given a 4-variable logic expression, simplify it using Entered Variable Map and realize the simplified logic expression using 8:1 multiplexer IC.
   b) Design and develop the Verilog /VHDL code for an 8:1 multiplexer. Simulate and verify its working.

6. a) Design and implement code converter I)Binary to Gray (II) Gray to Binary Code using basic gates.

7. Design and verify the Truth Table of 3-bit Parity Generator and 4-bit Parity Checker using basic Logic Gates with an even parity bit.

8. a) Realize a J-K Master / Slave Flip-Flop using NAND gates and verify its truth table.
   b) Design and develop the Verilog / VHDL code for D Flip-Flop with positive-edge triggering. Simulate and verify its working.

9. a) Design and implement a mod-n (n<8) synchronous up counter using J-K Flip-Flop ICs and demonstrate its working.
   b) Design and develop the Verilog / VHDL code for mod-8 up counter. Simulate and verify its working.

10. Design and implement an asynchronous counter using decade counter IC to count up from 0 to n (n<=9) and demonstrate on 7-segment display (using IC-7447).

11. Generate a Ramp output waveform using DAC0800 (Inputs are given to DAC through IC74393 dual 4-bit binary counter).

Study experiment

12. To study 4-bitALU using IC-74181.

Course outcomes:

On the completion of this laboratory course, the students will be able to:

- Use various Electronic Devices like Cathode ray Oscilloscope, Signal generators, Digital Trainer Kit, Multimeters and components like Resistors, Capacitors, Op amp and Integrated Circuit.
- Design and demonstrate various combinational logic circuits.
- Design and demonstrate various types of counters and Registers using Flip-flops
- Use simulation package to design circuits.
- Understand the working and implementation of ALU.

Graduate Attributes (as per NBA)

1. Engineering Knowledge
2. Problem Analysis
3. Design/Development of Solutions
4. Modern Tool Usage

Conduction of Practical Examination:

1. All laboratory experiments (1 to 11 nos) are to be included for practical examination.
2. Students are allowed to pick one experiment from the lot.
3. Strictly follow the instructions as printed on the cover page of answer script.
4. Marks distribution:
   a) For questions having part a only- Procedure + Conduction + Viva:**20 + 50 +10 =80 Marks**
   b) For questions having part a and b
   Part a- Procedure + Conduction + Viva:1**0 + 35 +05= 50 Marks**
   Part b- Procedure + Conduction + Viva:1**0 + 15 +05= 30 Marks**
5. **Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.**

DATA STRUCTURES LABORATORY

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER - III

Laboratory Code	**15CSL38**	**IA Marks**	**20**
Number of Lecture Hours/Week	**01I + 02P**	**Exam Marks**	**80**
Total Number of Lecture Hours	**40**	**Exam Hours**	**03**

CREDITS - 02

Course objectives:

This laboratory course enable students to get practical experience in design, develop, implement, analyze and evaluation/testing of

- Asymptotic performance of algorithms.
- Linear data structures and their applications such as Stacks, Queues and Lists
- Non-Linear Data Structures and their Applications such as Trees and Graphs
- Sorting and Searching Algorithms

Descriptions (if any)

Implement all the experiments in C Language under Linux / Windows environment.

Laboratory Experiments:

1. Design, Develop and Implement a menu driven Program in C for the following **Array** operations
    a. Creating an Array of **N** Integer Elements
    b. Display of Array Elements with Suitable Headings
    c. Inserting an Element (**ELEM**) at a given valid Position (**POS**)
    d. Deleting an Element at a given valid Position(**POS**)
    e. Exit.

    Support the program with functions for each of the above operations.

2. Design, Develop and Implement a Program in C for the following operationson **Strings**
    a. Read a main String (**STR),** a Pattern String (**PAT**) and a Replace String (**REP**)
    b. Perform Pattern Matching Operation: Find and Replace all occurrences of **PAT** in **STR** with **REP** if **PAT** exists in **STR.** Report suitable messages in case **PAT** does not exist in **STR**

    Support the program with functions for each of the above operations. Don't use Built-in functions.

3. Design, Develop and Implement a menu driven Program in C for the following operations on **STACK** of Integers (Array Implementation of Stack with maximum size **MAX**)
    a. ***Push*** an Element on to Stack
    b. ***Pop*** an Element from Stack
    c. Demonstrate how Stack can be used to check ***Palindrome***
    d. Demonstrate ***Overflow*** and ***Underflow*** situations on Stack

e. Display the status of Stack
f. Exit

Support the program with appropriate functions for each of the above operations

4. Design, Develop and Implement a Program in C for converting an Infix Expression to Postfix Expression. Program should support for both parenthesized and free parenthesized expressions with the operators: **+, -, *, /, %(Remainder), ^(Power)** and **alphanumeric** operands**.**

5. Design, Develop and Implement a Program in C for the following Stack Applications
   a. Evaluation of **Suffix expression** with single digit operands and operators: **+, -, *, /, %, ^**
   b. Solving **Tower of Hanoi** problem with **n** disks

6. Design, Develop and Implement a menu driven Program in C for the following operations on **Circular QUEUE** of Characters (Array Implementation of Queue with maximum size **MAX**)
   a. Insert an Element on to Circular QUEUE
   b. Delete an Element from Circular QUEUE
   c. Demonstrate ***Overflow*** and ***Underflow*** situations on Circular QUEUE
   d. Display the status of Circular QUEUE
   e. Exit

   Support the program with appropriate functions for each of the above operations

Continued:

7. Design, Develop and Implement a menu driven Program in C for the following operations on **Singly Linked List (SLL)** of Student Data with the fields: ***USN, Name, Branch, Sem, PhNo***
   a. Create a **SLL** of **N** Students Data by using ***front insertion***.
   b. Display the status of **SLL** and count the number of nodes in it
   c. Perform Insertion / Deletion at End of **SLL**
   d. Perform Insertion / Deletion at Front of **SLL(Demonstration of stack)**
   e. Exit

8. Design, Develop and Implement a menu driven Program in C for the following operations on **Doubly Linked List (DLL)** of Employee Data with the fields: ***SSN, Name, Dept, Designation, Sal, PhNo***
   a. Create a **DLL** of **N** Employees Data by using ***end insertion***.
   b. Display the status of **DLL** and count the number of nodes in it
   c. Perform Insertion and Deletion at End of **DLL**
   d. Perform Insertion and Deletion at Front of **DLL**
   e. Demonstrate how this **DLL** can be used as **Double Ended Queue**
   f. Exit

9. Design, Develop and Implement a Program in C for the following operationson **Singly Circular Linked List (SCLL)** with header nodes
    a. Represent and Evaluate a Polynomial **$P(x,y,z) = 6x^2y^2z-4yz^5+3x^3yz+2xy^5z-2xyz^3$**
    b. Find the sum of two polynomials **POLY1(x,y,z)** and **POLY2(x,y,z)** and store the result in **POLYSUM(x,y,z)**

    Support the program with appropriate functions for each of the above operations
10. Design, Develop and Implement a menu driven Program in C for the following operations on **Binary Search Tree (BST)** of Integers
    a. Create a BST of **N** Integers: 6, 9, 5, 2, 8, 15, 24, 14, 7, 8, 5, 2
    b. Traverse the BST in Inorder, Preorder and Post Order
    c. Search the BST for a given element (**KEY**) and report the appropriate message
    e. Exit

11. Design, Develop and Implement a Program in C for the following operations on **Graph(G)** of Cities
    a. Create a Graph of **N** cities using Adjacency Matrix.
    b. Print all the nodes **reachable** from a given starting node in a digraph using DFS/BFS method

12. Given a File of **N** employee records with a set **K** of Keys(4-digit) which uniquely determine the records in file **F**. Assume that file **F** is maintained in memory by a Hash Table(HT) of **m** memory locations with **L** as the set of memory addresses (2-digit) of locations in HT. Let the keys in **K** and addresses in **L** are Integers. Design and develop a Program in C that uses Hash function **H: K $\rightarrow$L** as H(**K**)=**K** mod **m (remainder** method), and implement hashing technique to map a given key **K** to the address space **L.** Resolve the collision (if any) using **linear probing**.

Course outcomes:

On the completion of this laboratory course, the students will be able to:

- Analyze and Compare various linear and non-linear data structures
- Code, debug and demonstrate the working nature of different types of data structures and their applications
- Implement, analyze and evaluate the searching and sorting algorithms
- Choose the appropriate data structure for solving real world problems

Graduate Attributes (as per NBA)

1. Engineering Knowledge
2. Problem Analysis
3. Design/Development of Solutions
4. Modern Tool Usage

Conduction of Practical Examination:

1. All laboratory experiments (**TWELVE** nos) are to be included for practical examination.
2. Students are allowed to pick one experiment from the lot.
3. Strictly follow the instructions as printed on the cover page of answer script
4. Marks distribution: Procedure + Conduction + Viva:**20 + 50 +10 (80)**
5. **Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.**

ENGINEERING MATHEMATICS-IV

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – IV

Subject Code	15MAT41	IA Marks	20
Number of Lecture Hours/Week	04	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03

CREDITS – 04

Course objectives: This course will enable students to

- Formulate, solve and analyze engineering problems.
- Apply numerical methods to solve ordinary differential equations.
- Apply finite difference method to solve partial differential equations.
- Perform complex analysis.
- Interpret use of sampling theory.
- Apply joint probability distribution and stochastic process.

Module 1	**Teaching Hours**
Numerical Methods: Numerical solution of ordinary differential equations of first order and first degree, Picard's method, Taylor's series method, modified Euler's method, Runge-Kutta method of fourth order. Milne's and Adams-Bashforth predictor and corrector methods (No derivations of formulae). Numerical solution of simultaneous first order ordinary differential equations, Picard's method, Runge-Kutta method of fourth order	**10 Hours**
Module 2	
Numerical Methods: Numerical solution of second order ordinary differential equations, Picard's method, Runge-Kutta method and Milne's method. **Special Functions:** Bessel's functions- basic properties, recurrence relations, orthogonality and generating functions. Legendre's functions - Legendre's polynomial, Rodrigue's formula, problems.	**10 Hours**
Module 3	
Complex Variables: Function of a complex variable, limits, continuity, differentiability,. Analytic functions-Cauchy-Riemann equations in Cartesian and polar forms. Properties and construction of analytic functions. Complex line integrals-Cauchy's theorem and Cauchy's integral formula, Residue, poles, Cauchy's Residue theorem with proof and problems. **Transformations:** Conformal transformations, discussion of transformations: = , = , = + (/) and bilinear transformations.	**10 Hours**
Module 4	
Probability Distributions: Random variables (discrete and continuous), probability functions. Poisson distributions**,** geometric distribution, uniform distribution, exponential and normal distributions, Problems. **Joint probability distribution:** Joint Probability distribution for two variables, expectation, covariance, correlation coefficient.	**10 Hours**
Module 5	
Sampling Theory: Sampling, Sampling distributions, standard error, test of hypothesis for means and proportions, confidence limits for means, student's t-distribution, Chi-square distribution as a test of goodness of fit. **Stochastic process:** Stochastic process, probability vector, stochastic matrices, fixed points, regular stochastic matrices, Markov chains, higher transition probability.	**10 Hours**

Course Outcomes: After studying this course, students will be able to:

- Use appropriate numerical methods to solve first and second order ordinary differential equations.
- Use Bessel's and Legendre's function which often arises when a problem possesses axial and spherical symmetry, such as in quantum mechanics, electromagnetic theory, hydrodynamics and heat conduction.
- State and prove Cauchy's theorem and its consequences including Cauchy's integral formula.
- Compute residues and apply the residue theorem to evaluate integrals.
- Analyze, interpret, and evaluate scientific hypotheses and theories using rigorous statistical methods.

Graduate Attributes

- Engineering Knowledge
- Problem Analysis
- Life-Long Learning
- Conduct Investigations of Complex Problems

Question paper pattern:

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. B.V.Ramana "Higher Engineering Mathematics" Tata McGraw-Hill, 2006.
2. B. S. Grewal," Higher Engineering Mathematics", Khanna publishers, 42nd edition, 2013.

Reference Books:

1. N P Bali and Manish Goyal, "A text book of Engineering mathematics" , Laxmi publications, latest edition.
2. Kreyszig, "Advanced Engineering Mathematics " - 9th edition, Wiley, 2013.
3. H. K Dass and Er. RajnishVerma, "Higher Engineering Mathematics", S. Chand, 1st ed, 2011.

SOFTWARE ENGINEERING

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – IV

Subject Code	15CS42	IA Marks	20
Number of Lecture Hours/Week	04	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03

CREDITS – 04

Course objectives: This course will enable students to

- Outline software engineering principles and activities involved in building large software programs.
- Identify ethical and professional issues and explain why they are of concern to software engineers.
- Describe the process of requirements gathering, requirements classification, requirements specification and requirements validation.
- Differentiate system models, use UML diagrams and apply design patterns.
- Discuss the distinctions between validation testing and defect testing.
- Recognize the importance of software maintenance and describe the intricacies involved in software evolution.
- Apply estimation techniques, schedule project activities and compute pricing.
- Identify software quality parameters and quantify software using measurements and metrics.
- List software quality standards and outline the practices involved.
- Recognize the need for agile software development, describe agile methods, apply agile practices and plan for agility.

Module 1	**Teaching Hours**
Introduction: Software Crisis, Need for Software Engineering. Professional Software Development, Software Engineering Ethics. Case Studies. **Software Processes**: Models: Waterfall Model **(Sec 2.1.1)**, Incremental Model **(Sec 2.1.2)** and Spiral Model **(Sec 2.1.3)**. Process activities. **Requirements Engineering**: Requirements Engineering Processes **(Chap 4)**. Requirements Elicitation and Analysis **(Sec 4.5).** Functional and non-functional requirements **(Sec 4.1)**. The software Requirements Document **(Sec 4.2)**. Requirements Specification **(Sec 4.3)**. Requirements validation **(Sec 4.6)**. Requirements Management **(Sec 4.7)**.	**12 Hours**
Module 2	
System Models: Context models **(Sec 5.1)**. Interaction models **(Sec 5.2)**. Structural models **(Sec 5.3)**. Behavioral models **(Sec 5.4)**. Model-driven engineering **(Sec 5.5)**. **Design and Implementation**: Introduction to RUP **(Sec 2.4)**, Design Principles **(Chap 17)**. Object-Oriented design using the UML **(Sec 7.1)**. Design patterns **(Sec 7.2)**. Implementation issues **(Sec 7.3)**. Open source development **(Sec 7.4)**.	**11 Hours**
Module 3	
Software Testing: Development testing **(Sec 8.1)**, Test-driven development **(Sec 8.2)**, Release testing **(Sec 8.3)**, User testing **(Sec 8.4)**. Test Automation **(Page no 42, 70,212, 231,444,695)**. **Software Evolution**: Evolution processes **(Sec 9.1)**. Program evolution dynamics **(Sec 9.2)**. Software maintenance **(Sec 9.3)**. Legacy system management **(Sec 9.4)**.	**9 Hours**

Module 4	
Project Planning: Software pricing **(Sec 23.1)**. Plan-driven development **(Sec 23.2)**. Project scheduling **(Sec 23.3)**: Estimation techniques **(Sec 23.5)**. **Quality management**: Software quality **(Sec 24.1)**. Reviews and inspections **(Sec 24.3)**. Software measurement and metrics **(Sec 24.4)**. Software standards **(Sec 24.2)**	**10 Hours**
Module 5	
Agile Software Development: Coping with Change **(Sec 2.3)**, The Agile Manifesto: Values and Principles. Agile methods: SCRUM (Ref "**The SCRUM Primer, Ver 2.0**") and Extreme Programming **(Sec 3.3)**. Plan-driven and agile development **(Sec 3.2)**. Agile project management **(Sec 3.4),** Scaling agile methods **(Sec 3.5)**:	**8 Hours**

Course Outcomes: After studying this course, students will be able to:

- Design a software system, component, or process to meet desired needs within realistic constraints.
- Assess professional and ethical responsibility
- Function on multi-disciplinary teams
- Use the techniques, skills, and modern engineering tools necessary for engineering practice
- Analyze, design, implement, verify, validate, implement, apply, and maintain software systems or parts of software systems.

Graduate Attributes

- Project Management and Finance
- Conduct Investigations of Complex Problems
- Modern Tool Usage
- Ethics

Question paper pattern:

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Ian Sommerville: Software Engineering, 9th Edition, Pearson Education, 2012.
(Listed topics only from Chapters 1,2,3,4, 5, 7, 8, 9, 23, and 24)
2. The SCRUM Primer, Ver 2.0, http://www.goodagile.com/scrumprimer/scrumprimer20.pdf

Reference Books:

1. Roger S. Pressman: Software Engineering-A Practitioners approach, 7th Edition, Tata McGraw Hill**.**
2. Pankaj Jalote: An Integrated Approach to Software Engineering, Wiley India

Web Reference for eBooks on Agile:

1. http://agilemanifesto.org/
2. http://www.jamesshore.com/Agile-Book/

DESIGN AND ANALYSIS OF ALGORITHMS

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – IV

Subject Code	15CS43	IA Marks	20
Number of Lecture Hours/Week	04	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03

CREDITS – 04

Course objectives: This course will enable students to

- Explain various computational problem solving techniques.
- Apply appropriate method to solve a given problem.
- Describe various methods of algorithm analysis.

Module 1	**Teaching Hours**
Introduction: What is an Algorithm? **(T2:1.1),** Algorithm Specification **(T2:1.2),** Analysis Framework **(T1:2.1), Performance Analysis**: Space complexity, Time complexity **(T2:1.3)**. **Asymptotic Notations:** Big-Oh notation (O), Omega notation (Ω), Theta notation (Θ), and Little-oh notation (o), Mathematical analysis of Non-Recursive and recursive Algorithms with Examples **(T1:2.2, 2.3, 2.4). Important Problem Types:** Sorting, Searching, String processing, Graph Problems, Combinatorial Problems. **Fundamental Data Structures:** Stacks, Queues, Graphs, Trees, Sets and Dictionaries. **(T1:1.3,1.4)**	**10 Hours**
Module 2	
Divide and Conquer: General method, Binary search, Recurrence equation for divide and conquer, Finding the maximum and minimum **(T2:3.1, 3.3, 3.4)**, Merge sort, Quick sort **(T1:4.1, 4.2)**, Strassen's matrix multiplication **(T2:3.8)**, Advantages and Disadvantages of divide and conquer. **Decrease and Conquer Approach:** Topological Sort. **(T1:5.3)**	**10 Hours**
Module 3	
Greedy Method: General method, Coin Change Problem, Knapsack Problem, Job sequencing with deadlines **(T2:4.1, 4.3, 4.5). Minimum cost spanning trees:** Prim's Algorithm, Kruskal's Algorithm **(T1:9.1, 9.2)**. **Single source shortest paths:** Dijkstra's Algorithm **(T1:9.3)**. **Optimal Tree problem:** Huffman Trees and Codes **(T1:9.4)**. **Transform and Conquer Approach:** Heaps and Heap Sort **(T1:6.4)**.	**10 Hours**
Module 4	
Dynamic Programming: General method with Examples, Multistage Graphs **(T2:5.1, 5.2)**. **Transitive Closure:** Warshall's Algorithm, **All Pairs Shortest Paths:** Floyd's Algorithm, Optimal Binary Search Trees, Knapsack problem **((T1:8.2, 8.3, 8.4)**, Bellman-Ford Algorithm **(T2:5.4)**, Travelling Sales Person problem **(T2:5.9)**, Reliability design **(T2:5.8)**.	**10 Hours**
Module 5	
Backtracking: General method **(T2:7.1)**, N-Queens problem **(T1:12.1)**, Sum of subsets problem **(T1:12.1)**, Graph coloring **(T2:7.4)**, Hamiltonian cycles **(T2:7.5)**. **Branch and Bound:** Assignment Problem, Travelling Sales Person problem **(T1:12.2)**, **0/1 Knapsack problem (T2:8.2, T1:12.2):** LC Branch and Bound solution **(T2:8.2)**, FIFO Branch and Bound solution **(T2:8.2)**. **NP-Complete and NP-Hard problems:** Basic	**10 Hours**

concepts, non-deterministic algorithms, P, NP, NP-Complete, and NP-Hard classes **(T2:11.1)**.	

Course Outcomes: After studying this course, students will be able to

- Describe computational solution to well known problems like searching, sorting etc.
- Estimate the computational complexity of different algorithms.
- Devise an algorithm using appropriate design strategies for problem solving.

Graduate Attributes

- Engineering Knowledge
- Problem Analysis
- Design/Development of Solutions
- Conduct Investigations of Complex Problems
- Life-Long Learning

Question paper pattern:

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

T1. Introduction to the Design and Analysis of Algorithms, Anany Levitin:, 2rd Edition, 2009. Pearson.
T2. Computer Algorithms/C++, Ellis Horowitz, Satraj Sahni and Rajasekaran, 2nd Edition, 2014, Universities Press

Reference Books:

1. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein, 3rd Edition, PHI
2. Design and Analysis of Algorithms , S. Sridhar, Oxford (Higher Education)

MICROPROCESSORS AND MICROCONTROLLERS

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – IV

Subject Code	15CS44	IA Marks	20
Number of Lecture Hours/Week	04	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03

CREDITS – 04

Course objectives: This course will enable students to

- Make familiar with importance and applications of microprocessors and microcontrollers
- Expose architecture of 8086 microprocessor and ARM processor
- Familiarize instruction set of ARM processor

Module 1	**Teaching Hours**
The x86 microprocessor: Brief history of the x86 family, Inside the 8088/86, Introduction to assembly programming, Introduction to Program Segments, The Stack, Flag register, x86 Addressing Modes. **Assembly language programming:** Directives & a Sample Program, Assemble, Link & Run a program, More Sample programs, Control Transfer Instructions, Data Types and Data Definition, Full Segment Definition, Flowcharts and Pseudo code. **Text book 1: Ch 1: 1.1 to 1.7, Ch 2: 2.1 to 2.7**	**10 Hours**
Module 2	
x86: Instructions sets description**, Arithmetic and logic instructions and programs:** Unsigned Addition and Subtraction, Unsigned Multiplication and Division, Logic Instructions, BCD and ASCII conversion, Rotate Instructions. **INT 21H and INT 10H Programming :** Bios INT 10H Programming , DOS Interrupt 21H. 8088/86 Interrupts, x86 PC and Interrupt Assignment. **Text book 1: Ch 3: 3.1 to 3.5, Ch 4: 4.1 , 4.2 Chapter 14: 14.1 and 14.2**	**10 Hours**
Module 3	
Signed Numbers and Strings: Signed number Arithmetic Operations, String operations. **Memory and Memory interfacing: M**emory address decoding, data integrity in RAM and ROM, 16-bit memory interfacing. **8255 I/O programming:** I/O addresses MAP of x86 PC's, programming and interfacing the 8255. **Text book 1: Ch 6: 6.1, 6.2. Ch 10: 10.2, 10.4, 10.5. Ch 11: 11.1 to 11.4**	**10 Hours**
Module 4	
Microprocessors versus Microcontrollers, **ARM Embedded Systems :**The RISC design philosophy, The ARM Design Philosophy, Embedded System Hardware, Embedded System Software, **ARM Processor Fundamentals :** Registers , Current Program Status Register , Pipeline, Exceptions, Interrupts, and the Vector Table , Core Extensions **Text book 2:Ch 1:1.1 to 1.4, Ch 2:2.1 to 2.5**	**10 Hours**
Module 5	
Introduction to the ARM Instruction Set : Data Processing Instructions , Branch Instructions, Software Interrupt Instructions, Program Status Register Instructions, Coprocessor Instructions, Loading Constants, Simple programming exercises. **Text book 2: Ch 3:3.1 to 3.6 (Excluding 3.5.2)**	**10 Hours**

Course Outcomes: After studying this course, students will be able to

- Differentiate between microprocessors and microcontrollers
- Design and develop assembly language code to solve problems
- Gain the knowledge for interfacing various devices to x86 family and ARM processor
- Demonstrate design of interrupt routines for interfacing devices

Graduate Attributes

- Engineering Knowledge
- Problem Analysis
- Design/Development of Solutions

Question paper pattern:

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Muhammad Ali Mazidi, Janice Gillispie Mazidi, Danny Causey, The x86 PC Assembly Language Design and Interfacing, 5th Edition, Pearson, 2013.
2. **ARM system developers guide**, Andrew N Sloss, Dominic Symes and Chris Wright, Elsevier,Morgan Kaufman publishers, 2008.

Reference Books:

1. Douglas V. Hall: Microprocessors and Interfacing, Revised 2nd Edition, TMH, 2006.
2. K. Udaya Kumar & B.S. Umashankar : Advanced Microprocessors & IBM-PC Assembly Language Programming, TMH 2003.
3. Ayala : The 8086 Microprocessor: programming and interfacing - 1st edition, Cengage Learning
4. The Definitive Guide to the ARM Cortex-M3, by Joseph Yiu, 2nd Edition , Newnes, 2009
5. The Insider's Guide to the ARM7 based microcontrollers, Hitex Ltd.,1st edition, 2005
6. ARM System-on-Chip Architecture, Steve Furber, Second Edition, Pearson, 2015
7. Architecture, Programming and Interfacing of Low power Processors- ARM7, Cortex-M and MSP430, Lyla B Das Cengage Learning, 1st Edition

OBJECT ORIENTED CONCEPTS

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – IV

Subject Code	15CS45	IA Marks	20
Number of Lecture Hours/Week	04	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03

CREDITS – 04

Course objectives: This course will enable students to

- Learn fundamental features of object oriented language and JAVA
- Set up Java JDK environment to create, debug and run simple Java programs.
- Create multi-threaded programs and event handling mechanisms.
- Introduce event driven Graphical User Interface (GUI) programming using applets and swings**.**

Module 1	**Teaching Hours**
Introduction to Object Oriented Concepts: A Review of structures, Procedure–Oriented Programming system, Object Oriented Programming System, Comparison of Object Oriented Language with C, Console I/O, variables and reference variables, Function Prototyping, Function Overloading. **Class and Objects:** Introduction, member functions and data, objects and functions, objects and arrays, Namespaces, Nested classes, Constructors, Destructors. **Text book 1: Ch 1: 1.1 to 1.9 Ch 2: 2.1 to 2.6 Ch 4: 4.1 to 4.2**	**10 Hours**
Module 2	
Introduction to Java: Java's magic: the Byte code; Java Development Kit (JDK); the Java Buzzwords, Object-oriented programming; Simple Java programs. Data types, variables and arrays, Operators, Control Statements. **Text book 2: Ch:1 Ch: 2 Ch:3 Ch:4 Ch:5**	**10 Hours**
Module 3	
Classes, Inheritance, Exceptions, Packages and Interfaces: Classes: Classes fundamentals; Declaring objects; Constructors, this keyword, garbage collection. **Inheritance:** inheritance basics, using super, creating multi level hierarchy, method overriding. **Exception handling:** Exception handling in Java. Packages, Access Protection, Importing Packages, Interfaces. **Text book 2: Ch:6 Ch: 8 Ch:9 Ch:10**	**10 Hours**
Module 4	
Multi Threaded Programming, Event Handling: Multi Threaded Programming: What are threads? How to make the classes threadable ; Extending threads; Implementing runnable; Synchronization; Changing state of the thread; Bounded buffer problems, read-write problem, producer consumer problems. **Event Handling:** Two event handling mechanisms; The delegation event model; Event classes; Sources of events; Event listener interfaces; Using the delegation event model; Adapter classes; Inner classes. **Text book 2: Ch 11: Ch: 22**	**10 Hours**
Module 5	
The Applet Class: Introduction, Two types of Applets; Applet basics; Applet Architecture; An Applet skeleton; Simple Applet display methods; Requesting repainting;	**10 Hours**

Using the Status Window; The HTML APPLET tag; Passing parameters to Applets; getDocumentbase() and getCodebase(); ApletContext and showDocument(); The AudioClip Interface; The AppletStub Interface;Output to the Console. **Swings:** Swings: The origins of Swing; Two key Swing features; Components and Containers; The Swing Packages; A simple Swing Application; Create a Swing Applet; Jlabel and ImageIcon; JTextField;The Swing Buttons; JTabbedpane; JScrollPane; JList; JComboBox; JTable. **Text book 2: Ch 21: Ch: 29 Ch: 30**	

Course Outcomes: After studying this course, students will be able to

- Explain the object-oriented concepts and JAVA.
- Develop computer programs to solve real world problems in Java.
- Develop simple GUI interfaces for a computer program to interact with users, and to understand the event-based GUI handling principles using Applets and swings.

Graduate Attributes

- Programming Knowledge
- Design/Development of Solutions
- Conduct Investigations of Complex Problems
- Life-Long Learning

Question paper pattern:

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Sourav Sahay, Object Oriented Programming with C++ , 2nd Ed, Oxford University Press,2006
   (Chapters 1, 2, 4)
2. Herbert Schildt, Java The Complete Reference, 7th Edition, Tata McGraw Hill, 2007.
   (Chapters 1, 2, 3, 4, 5, 6, 8, 9,10, 11, 21, 22, 29, 30)

Reference Book:

1. Mahesh Bhave and Sunil Patekar, "Programming with Java", First Edition, Pearson Education,2008, ISBN:9788131720806
2. Herbert Schildt, The Complete Reference C++, 4th Edition, Tata McGraw Hill, 2003.
3. Stanley B.Lippmann, Josee Lajore, C++ Primer, 4th Edition, Pearson Education, 2005.
4. Rajkumar Buyya,S Thamarasi selvi, xingchen chu, Object oriented Programming with java, Tata McGraw Hill education private limited.
5. Richard A Johnson, Introduction to Java Programming and OOAD, CENGAGE Learning.
6. E Balagurusamy, Programming with Java A primer, Tata McGraw Hill companies.

Note: Every institute shall organize a bridge organize on C++ either in the vacation or in the beginning of even semester.

DATA COMMUNICATION

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – IV

Subject Code	15CS46	IA Marks	20
Number of Lecture Hours/Week	04	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03

CREDITS – 04

Course objectives: This course will enable students to

- Comprehend the transmission technique of digital data between two or more computers and a computer network that allows computers to exchange data.
- Explain with the basics of data communication and various types of computer networks;
- Illustrate TCP/IP protocol suite and switching criteria.
- Demonstrate Medium Access Control protocols for reliable and noisy channels.
- Expose wireless and wired LANs along with IP version.

Contents	**Teaching Hours**
Module 1	
Introduction: Data Communications, Networks, Network Types, Internet History, Standards and Administration, **Networks Models**: Protocol Layering, TCP/IP Protocol suite, The OSI model, **Introduction to Physical Layer-1:** Data and Signals**,** Digital Signals, Transmission Impairment, Data Rate limits, Performance, **Digital Transmission**: Digital to digital conversion (Only Line coding: Polar, Bipolar and Manchester coding).	**10 Hours**
Module 2	
Physical Layer-2: Analog to digital conversion (only PCM), Transmission Modes, **Analog Transmission**: Digital to analog conversion, **Bandwidth Utilization**: Multiplexing and Spread Spectrum, **Switching**: Introduction, Circuit Switched Networks and Packet switching.	**10 Hours**
Module 3	
Error Detection and Correction: Introduction, Block coding, Cyclic codes, Checksum, Forward error correction**, Data link control**: DLC services, Data link layer protocols, HDLC, and Point to Point protocol (Framing, Transition phases only).	**10 Hours**
Module 4	
Media Access control: Random Access, Controlled Access and Channelization, **Wired LANs Ethernet**: Ethernet Protocol, Standard Ethernet, Fast Ethernet, Gigabit Ethernet and 10 Gigabit Ethernet, **Wireless LANs:** Introduction, IEEE 802.11 Project and Bluetooth.	**10 Hours**
Module 5	
Other wireless Networks: WIMAX, Cellular Telephony, Satellite networks, **Network layer Protocols** : Internet Protocol, ICMPv4,Mobile IP, **Next generation IP**: IPv6 addressing, The IPv6 Protocol, The ICMPv6 Protocol and Transition from IPv4 to IPv6.	**10 Hours**

Course Outcomes: After studying this course, students will be able to

- Illustrate basic computer network technology.
- Identify the different types of network topologies and protocols.
- Enumerate the layers of the OSI model and TCP/IP functions of each layer.
- Make out the different types of network devices and their functions within a network

- Demonstrate the skills of subnetting and routing mechanisms.

Graduate Attributes

1. Engineering Knowledge
2. Design Development of solution(Partly)
3. Modern Tool Usage
4. Problem Analysis

Question paper pattern:

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Book:

Behrouz A. Forouzan, Data Communications and Networking 5E, 5th Edition, Tata McGraw-Hill, 2013. (Chapters 1.1 to 1.5, 2.1 to 2.3, 3.1, 3.3 to 3.6, 4.1 to 4.3, 5.1, 6.1, 6.2, 8.1 to 8.3, 10.1 to 10.5, 11.1 to 11.4, 12.1 to 12.3, 13.1 to 13.5, 15.1 to 15.3, 16.1 to 16.3, 19.1 to 19.3, 22.1 to 22.4)

Reference Books:

1. Alberto Leon-Garcia and Indra Widjaja: Communication Networks - Fundamental Concepts and Key architectures, 2nd Edition Tata McGraw-Hill, 2004.
2. William Stallings: Data and Computer Communication, 8th Edition, Pearson Education, 2007.
3. Larry L. Peterson and Bruce S. Davie: Computer Networks – A Systems Approach, 4th Edition, Elsevier, 2007.
4. Nader F. Mir: Computer and Communication Networks, Pearson Education, 2007

DESIGN AND ANALYSIS OF ALGORITHM LABORATORY
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – IV

Subject Code	15CSL47	IA Marks	20
Number of Lecture Hours/Week	01 I + 02 P	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 02

Course objectives: This course will enable students to

- Design and implement various algorithms in JAVA
- Employ various design strategies for problem solving.
- Measure and compare the performance of different algorithms.

Description

Design, develop, and implement the specified algorithms for the following problems using Java language under LINUX /Windows environment.Netbeans/Eclipse IDE tool can be used for development and demonstration.

Experiments

1	A	Create a Java class called ***Student***with the following details as variables within it. (i) USN (ii) Name (iii) Branch (iv) Phone Write a Java program to create *nStudent* objects and print the USN, Name, Branch, and Phoneof these objects with suitable headings.
	B	Write a Java program to implement the Stack using arrays. Write Push(), Pop(), and Display() methods to demonstrate its working.
2	A	Design a superclass called ***Staff*** with details as StaffId, Name, Phone, Salary. Extend this class by writing three subclasses namely ***Teaching*** (domain, publications), ***Technical*** (skills), and ***Contract*** (period). Write a Java program to read and display at least 3 *staff* objects of all three categories.
	B	Write a Java class called ***Customer*** to store their name and date_of_birth. The date_of_birth format should be dd/mm/yyyy. Write methods to read customer data as <name, dd/mm/yyyy> and display as <name, dd, mm, yyyy> using StringTokenizer class considering the delimiter character as "/".
3	A	Write a Java program to read two integers *a* and*b*. Compute *a*/*b* and print, when *b* is not zero. Raise an exception when *b* is equal to zero.
	B	Write a Java program that implements a multi-thread application that has three threads. First thread generates a random integer for every 1 second; second thread computes the square of the number andprints; third thread will print the value of cube of the number.
4		Sort a given set of ***n*** integer elements using **Quick Sort** method and compute its time complexity. Run the program for varied values of *n*> 5000 and record the time taken to sort. Plot a graph of the time taken versus ***n***on graph sheet. The elements can be read from a file or can be generated using the random number generator. Demonstrate using Java how the divide-and-conquer method works along with its time complexity analysis: worst case, average case and best case.

5	Sort a given set of ***n*** integer elements using **Merge Sort** method and compute its time complexity. Run the program for varied values of ***n***> 5000, and record the time taken to sort. Plot a graph of the time taken versus ***n***on graph sheet. The elements can be read from a file or can be generated using the random number generator. Demonstrate using Java how the divide-and-conquer method works along with its time complexity analysis: worst case, average case and best case.
6	Implement in Java, the **0/1 Knapsack** problem using (a) Dynamic Programming method (b) Greedy method.
7	From a given vertex in a weighted connected graph, find shortest paths to other vertices using **Dijkstra's algorithm**. Write the program in Java.
8	Find Minimum Cost Spanning Tree of a given connected undirected graph using **Kruskal'salgorithm.** Use Union-Find algorithms in your program.
9	Find Minimum Cost Spanning Tree of a given connected undirected graph using **Prim's algorithm**.
10	Write Java programs to (a) Implement All-Pairs Shortest Paths problem using **Floyd's algorithm**. (b) Implement **Travelling Sales Person problem** using Dynamic programming.
11	Design and implement in Java to find a **subset** of a given set **S** = {Sl, S2,.....,Sn} of ***n*** positive integers whose SUM is equal to a given positive integer ***d***. For example, if S ={1, 2, 5, 6, 8} and ***d***= 9, there are two solutions {1,2,6}and {1,8}. Display a suitable message, if the given problem instance doesn't have a solution.
12	Design and implement in Java to find all **Hamiltonian Cycles** in a connected undirected Graph G of ***n*** vertices using backtracking principle.

Course Outcomes: The students should be able to:

- Design algorithms using appropriate design techniques (brute-force, greedy, dynamic programming, etc.)
- Implement a variety of algorithms such assorting, graph related, combinatorial, etc., in a high level language.
- Analyze and compare the performance of algorithms using language features.
- Apply and implement learned algorithm design techniques and data structuresto solve real-world problems.

Graduate Attributes

- Engineering Knowledge
- Problem Analysis
- Modern Tool Usage
- Conduct Investigations of Complex Problems
- Design/Development of Solutions

Conduction of Practical Examination:

All laboratory experiments (Twelve problems) are to be included for practical examination. Students are allowed to pick one experiment from the lot.
To generate the data set use random number generator function.
Strictly follow the instructions as printed on the cover page of answer script for breakup of marks
Marks distribution: Procedure + Conduction + Viva: 20 + 50 + 10 (80). Change of experiment is allowed only once and marks allotted to the procedure

MICROPROCESSOR AND MICROCONTROLLER LABORATORY

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – IV

Subject Code	15CSL48	IA Marks	20
Number of Lecture Hours/Week	01 I + 02 P	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 02

Course objectives: This course will enable students to

- To provide practical exposure to the students on microprocessors, design and coding knowledge on 80x86 family/ARM. To give the knowledge and practical exposure on connectivity and execute of interfacing devices with 8086/ARM kit like LED displays, Keyboards, DAC/ADC, and various other devices.

Description

Demonstration and Explanation hardware components and Faculty in-charge should explain 8086 architecture, pin diagram in one slot. The second slot, the Faculty in-charge should explain instruction set types/category etc. Students have to prepare a write-up on the same and include it in the Lab record and to be evaluated.

Laboratory Session-1: Write-up on Microprocessors, 8086 Functional block diagram, Pin diagram and description. The same information is also taught in theory class; this helps the students to understand better.

Laboratory Session-2: Write-up on Instruction group, Timing diagrams, etc. The same information is also taught in theory class; this helps the students to understand better.

Note: These TWO Laboratory sessions are used to fill the gap between theory classes and practical sessions. Both sessions are evaluated as lab experiments for 20 marks.

Experiments

- Develop and execute the following programs using 8086 Assembly Language. Any suitable assembler like MASM/TASM/8086 kit or any equivalent software may be used.
- Program should have suitable comments.
- The board layout and the circuit diagram of the interface are to be provided to the student during the examination.
- Software Required: Open source ARM Development platform, KEIL IDE and Proteus for simulation

SOFTWARE PROGRAMS: PART A

1. Design and develop an assembly language program to search a key element “X” in a list of ‘n’ 16-bit numbers. Adopt Binary search algorithm in your program for searching.
2. Design and develop an assembly program to sort a given set of ‘n’ 16-bit numbers in ascending order. Adopt Bubble Sort algorithm to sort given elements.
3. Develop an assembly language program to reverse a given string and verify whether it is a palindrome or not. Display the appropriate message.
4. Develop an assembly language program to compute nCr using recursive procedure. Assume that ‘n’ and ‘r’ are non-negative integers.

5. Design and develop an assembly language program to read the current time and Date from the system and display it in the standard format on the screen.
6. To write and simulate ARM assembly language programs for data transfer, arithmetic and logical operations (Demonstrate with the help of a suitable program).
7. To write and simulate C Programs for ARM microprocessor using KEIL (Demonstrate with the help of a suitable program)

Note : To use KEIL one may refer the book: Insider's Guide to the ARM7 based microcontrollers, Hitex Ltd.,1st edition, 2005

HARDWARE PROGRAMS: PART B

8. a. Design and develop an assembly program to demonstrate BCD Up-Down Counter (00-99) on the Logic Controller Interface.
   b. Design and develop an assembly program to read the status of two 8-bit inputs (X & Y) from the Logic Controller Interface and display X*Y.
9. Design and develop an assembly program to display messages "FIRE" and "HELP" alternately with flickering effects on a 7-segment display interface for a suitable period of time. Ensure a flashing rate that makes it easy to read both the messages (Examiner does not specify these delay values nor is it necessary for the student to compute these values).
10. Design and develop an assembly program to drive a Stepper Motor interface and rotate the motor in specified direction (clockwise or counter-clockwise) by N steps (Direction and N are specified by the examiner). Introduce suitable delay between successive steps. (Any arbitrary value for the delay may be assumed by the student).
11. Design and develop an assembly language program to
    a. Generate the Sine Wave using DAC interface (The output of the DAC is to be displayed on the CRO).
    b. Generate a Half Rectified Sine waveform using the DAC interface. (The output of the DAC is to be displayed on the CRO).
12. To interface LCD with ARM processor-- ARM7TDMI/LPC2148. Write and execute programs in C language for displaying text messages and numbers on LCD
13. To interface Stepper motor with ARM processor-- ARM7TDMI/LPC2148. Write a program to rotate stepper motor

Study Experiments:

1. Interfacing of temperature sensor with ARM freedom board (or any other ARM microprocessor board) and display temperature on LCD
2. To design ARM cortex based automatic number plate recognition system
3. To design ARM based power saving system

Course Outcomes: After studying this course, students will be able to

- Learn 80x86 instruction sets and gins the knowledge of how assembly language works.
- Design and implement programs written in 80x86 assembly language
- Know functioning of hardware devices and interfacing them to x86 family
- Choose processors for various kinds of applications.

Graduate Attributes

- Engineering Knowledge
- Problem Analysis
- Modern Tool Usage
- Conduct Investigations of Complex Problems
- Design/Development of Solutions

Conduction of Practical Examination:

- All laboratory experiments (all 7 + 6 nos) are to be included for practical examination.
- Students are allowed to pick one experiment from each of the lot.
- Strictly follow the instructions as printed on the cover page of answer script for breakup of marks
- PART –A: Procedure + Conduction + Viva: 10 + 25 +05 (40)
- PART –B: Procedure + Conduction + Viva: 10 + 25 +05 (40)
- Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.

PROGRAMMING IN JAVA
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – V

Subject Code	15CS561	IA Marks	20
Number of Lecture Hours/Week	3	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 03

Course objectives: This course will enable students to

- Learn fundamental features of object oriented language and JAVA
- Set up Java JDK environment to create, debug and run simple Java programs.
- Learn object oriented concepts using programming examples.
- Study the concepts of importing of packages and exception handling mechanism.
- Discuss the String Handling examples with Object Oriented concepts.

Module – 1	**Teaching Hours**
An Overview of Java: Object-Oriented Programming, A First Simple Program, A Second Short Program, Two Control Statements, Using Blocks of Code, Lexical Issues, The Java Class Libraries, Data Types, Variables, and Arrays: Java Is a Strongly Typed Language, The Primitive Types, Integers, Floating-Point Types, Characters, Booleans, A Closer Look at Literals, Variables, Type Conversion and Casting, Automatic Type Promotion in Expressions, Arrays, A Few Words About Strings **Text book 1: Ch 2, Ch 3**	**8 Hours**
Module – 2	
Operators: Arithmetic Operators, The Bitwise Operators, Relational Operators, Boolean Logical Operators, The Assignment Operator, The ? Operator, Operator Precedence, Using Parentheses, Control Statements: Java's Selection Statements, Iteration Statements, Jump Statements. **Text book 1: Ch 4, Ch 5**	**8 Hours**
Module – 3	
Introducing Classes: Class Fundamentals, Declaring Objects, Assigning Object Reference Variables, Introducing Methods, Constructors, The this Keyword, Garbage Collection, The finalize( ) Method, A Stack Class, A Closer Look at Methods and Classes: Overloading Methods, Using Objects as Parameters, A Closer Look at Argument Passing, Returning Objects, Recursion, Introducing Access Control, Understanding static, Introducing final, Arrays Revisited, Inheritance: Inheritance, Using super, Creating a Multilevel Hierarchy, When Constructors Are Called, Method Overriding, Dynamic Method Dispatch, Using Abstract Classes, Using final with Inheritance, The Object Class. **Text book 1: Ch 6, Ch 7.1-7.9, Ch 8.**	**8 Hours**
Module – 4	
Packages and Interfaces: Packages, Access Protection, Importing Packages, Interfaces, Exception Handling: Exception-Handling Fundamentals, Exception Types, Uncaught Exceptions, Using try and catch, Multiple catch Clauses, Nested try Statements, throw, throws, finally, Java's Built-in Exceptions, Creating Your Own Exception Subclasses, Chained Exceptions, Using Exceptions. **Text book 1: Ch 9, Ch 10**	**8 Hours**

Module – 5	
Enumerations, Type Wrappers, I/O, Applets, and Other Topics: I/O Basics, Reading Console Input, Writing Console Output, The PrintWriter Class, Reading and Writing Files, Applet Fundamentals, The transient and volatile Modifiers, Using instanceof, strictfp, Native Methods, Using assert, Static Import, Invoking Overloaded Constructors Through this( ), String Handling: The String Constructors, String Length, Special String Operations, Character Extraction, String Comparison, Searching Strings, Modifying a String, Data Conversion Using valueOf( ), Changing the Case of Characters Within a String , Additional String Methods, StringBuffer, StringBuilder. **Text book 1: Ch 12.1,12.2, Ch 13, Ch 15**	**8 Hours**
Course outcomes: The students should be able to:	
• Explain the object-oriented concepts and JAVA. • Develop computer programs to solve real world problems in Java. • Develop simple GUI interfaces for a computer program to interact with users	
Question paper pattern: The question paper will have TEN questions. There will be TWO questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer FIVE full questions, selecting ONE full question from each module.	
Text Books:	
1. Herbert Schildt, Java The Complete Reference, 7th Edition, Tata McGraw Hill, 2007. (Chapters 2, 3, 4, 5, 6,7, 8, 9,10, 12,13,15)	
Reference Books:	
1. Mahesh Bhave and Sunil Patekar, "Programming with Java", First Edition, Pearson Education,2008, ISBN:9788131720806. 2. Rajkumar Buyya,S Thamarasi selvi, xingchen chu, Object oriented Programming with java, Tata McGraw Hill education private limited. 3. E Balagurusamy, Programming with Java A primer, Tata McGraw Hill companies. 4. Anita Seth and B L Juneja, JAVA One step Ahead, Oxford University Press, 2017.	

ARTIFICIAL INTELLIGENCE
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – V

Subject Code	15CS562	IA Marks	20
Number of Lecture Hours/Week	3	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 03

Course objectives: This course will enable students to

- Identify the problems where AI is required and the different methods available
- Compare and contrast different AI techniques available.
- Define and explain learning algorithms

Module – 1	**Teaching Hours**
What is artificial intelligence?, Problems, Problem Spaces and search, Heuristic search technique **TextBook1: Ch 1, 2 and 3**	**8 Hours**
Module – 2	
Knowledge Representation Issues, Using Predicate Logic, Representing knowledge using Rules, **TextBoook1: Ch 4, 5 and 6.**	**8 Hours**
Module – 3	
Symbolic Reasoning under Uncertainty, Statistical reasoning, Weak Slot and Filter Structures. **TextBoook1: Ch 7, 8 and 9.**	**8 Hours**
Module – 4	
Strong slot-and-filler structures, Game Playing. **TextBoook1: Ch 10 and 12**	**8 Hours**
Module – 5	
Natural Language Processing, Learning, Expert Systems. **TextBook1: Ch 15,17 and 20**	**8 Hours**

Course outcomes: The students should be able to:

- Identify the AI based problems
- Apply techniques to solve the AI problems
- Define learning and explain various learning techniques
- Discuss on expert systems

Question paper pattern:
The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

Text Books:

1. E. Rich , K. Knight & S. B. Nair - Artificial Intelligence, 3/e, McGraw Hill.

Reference Books:

1. Artificial Intelligence: A Modern Approach, Stuart Rusell, Peter Norving, Pearson Education 2nd Edition.

1. Dan W. Patterson, Introduction to Artificial Intelligence and Expert Systems – Prentice Hal of India.
2. G. Luger, “Artificial Intelligence: Structures and Strategies for complex problem Solving”, Fourth Edition, Pearson Education, 2002.
3. Artificial Intelligence and Expert Systems Development by D W Rolston-Mc Graw hill.
4. N.P. Padhy “Artificial Intelligence and Intelligent Systems” , Oxford University Press-2015

EMBEDDED SYSTEMS
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – V

Subject Code	15CS563	IA Marks	20
Number of Lecture Hours/Week	3	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 03

Course objectives: This course will enable students to

- Provide a general overview of Embedded Systems
- Show current statistics of Embedded Systems
- Design, code, compile, and test real-time software
- Integrate a fully functional system including hardware and software.

Module – 1	**Teaching Hours**
Introduction to embedded systems: Embedded systems, Processor embedded into a system, Embedded hardware units and device in a system, Embedded software in a system, Examples of embedded systems, Design process in embedded system, Formalization of system design, Design process and design examples, Classification of embedded systems, skills required for an embedded system designer.	**8 Hours**
Module – 2	
Devices and communication buses for devices network: IO types and example, Serial communication devices, Parallel device ports, Sophisticated interfacing features in device ports, Wireless devices, Timer and counting devices, Watchdog timer, Real time clock, Networked embedded systems, Serial bus communication protocols, Parallel bus device protocols-parallel communication internet using ISA, PCI, PCI-X and advanced buses, Internet enabled systems-network protocols, Wireless and mobile system protocols.	**8 Hours**
Module – 3	
Device drivers and interrupts and service mechanism: Programming-I/O busy-wait approach without interrupt service mechanism, ISR concept, Interrupt sources, Interrupt servicing (Handling) Mechanism, Multiple interrupts, Context and the periods for context switching, interrupt latency and deadline, Classification of processors interrupt service mechanism from Context-saving angle, Direct memory access, Device driver programming.	**8 Hours**
Module – 4	
Inter process communication and synchronization of processes, Threads and tasks: Multiple process in an application, Multiple threads in an application, Tasks, Task states, Task and Data, Clear-cut distinction between functions. ISRS and tasks by their characteristics, concept and semaphores, Shared data, Inter-process communication, Signal function, Semaphore functions, Message Queue functions, Mailbox functions, Pipe functions, Socket functions, RPC functions.	**8 Hours**
Module – 5	
Real-time operating systems: OS Services, Process management, Timer functions, Event functions, Memory management, Device, file and IO subsystems management, Interrupt routines in RTOS environment and handling of interrupt source calls, Real-time operating systems, Basic design using an RTOS, RTOS task scheduling models, interrupt latency and response of the tasks	**8 Hours**

as performance metrics, OS security issues. Introduction to embedded software development process and tools, Host and target machines, Linking and location software.	

Course outcomes: The students should be able to:

- Distinguish the characteristics of embedded computer systems.
- Examine the various vulnerabilities of embedded computer systems.
- Design and develop modules using RTOS.
- Implement RPC, threads and tasks

Question paper pattern:

The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

Text Books:

1. Raj Kamal, "Embedded Systems: Architecture, Programming, and Design" 2nd / 3rd edition , Tata McGraw hill-2013.

Reference Books:

1. Marilyn Wolf, "Computer as Components, Principles of Embedded Computing System Design" 3rd edition, Elsevier-2014.

DOT NET FRAMEWORK FOR APPLICATION DEVELOPMENT
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – V

Subject Code	15CS564	IA Marks	20
Number of Lecture Hours/Week	3	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 03

Course objectives: This course will enable students to

- Inspect Visual Studio programming environment and toolset designed to build applications for Microsoft Windows
- Understand Object Oriented Programming concepts in C# programming language.
- Interpret Interfaces and define custom interfaces for application.
- Build custom collections and generics in C#
- Construct events and query data using query expressions

Module – 1	**Teaching Hours**
Introducing Microsoft Visual C# and Microsoft Visual Studio 2015: Welcome to C#, Working with variables, operators and expressions, Writing methods and applying scope, Using decision statements, Using compound assignment and iteration statements, Managing errors and exceptions **T1: Chapter 1 – Chapter 6**	**8 Hours**
Module – 2	
Understanding the C# object model: Creating and Managing classes and objects, Understanding values and references, Creating value types with enumerations and structures, Using arrays **Textbook 1: Ch 7 to 10**	**8 Hours**
Module – 3	
Understanding parameter arrays, Working with inheritance, Creating interfaces and defining abstract classes, Using garbage collection and resource management **Textbook 1: Ch 11 to 14**	**8 Hours**
Module – 4	
Defining Extensible Types with C#: Implementing properties to access fields, Using indexers, Introducing generics, Using collections **Textbook 1: Ch 15 to 18**	**8 Hours**
Module – 5	
Enumerating Collections, Decoupling application logic and handling events, Querying in-memory data by using query expressions, Operator overloading **Textbook 1: Ch 19 to 22**	**8 Hours**

Course outcomes: The students should be able to:

- Build applications on Visual Studio .NET platform by understanding the syntax and semantics of C#
- Demonstrate Object Oriented Programming concepts in C# programming language
- Design custom interfaces for applications and leverage the available built-in interfaces in building complex applications.
- Illustrate the use of generics and collections in C#
- Compose queries to query in-memory data and define own operator behaviour

Question paper pattern:

The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

Text Books:

1. John Sharp, Microsoft Visual C# Step by Step, 8th Edition, PHI Learning Pvt. Ltd. 2016

Reference Books:

1. Christian Nagel, "C# 6 and .NET Core 1.0", 1st Edition, Wiley India Pvt Ltd, 2016. Andrew Stellman and Jennifer Greene, "Head First C#", 3rd Edition, O'Reilly Publications, 2013.
2. Mark Michaelis, "Essential C# 6.0", 5th Edition, Pearson Education India, 2016.
3. Andrew Troelsen, "Prof C# 5.0 and the .NET 4.5 Framework", 6th Edition, Apress and Dreamtech Press, 2012.

CLOUD COMPUTING
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – V

Subject Code	15CS565	IA Marks	20
Number of Lecture Hours/Week	3	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 03

Course objectives: This course will enable students to

- Explain the technology and principles involved in building a cloud environment.
- Contrast various programming models used in cloud computing
- Choose appropriate cloud model for a given application

Module – 1	**Teaching Hours**
Introduction ,Cloud Computing at a Glance, The Vision of Cloud Computing, Defining a Cloud, A Closer Look, Cloud Computing Reference Model, Characteristics and Benefits, Challenges Ahead, Historical Developments, Distributed Systems, Virtualization, Web 2.0, Service-Oriented Computing, Utility-Oriented Computing, Building Cloud Computing Environments, Application Development, Infrastructure and System Development, Computing Platforms and Technologies, Amazon Web Services (AWS), Google AppEngine, Microsoft Azure, Hadoop, Force.com and Salesforce.com, Manjrasoft Aneka Virtualization, Introduction, Characteristics of Virtualized, Environments Taxonomy of Virtualization Techniques, Execution Virtualization, Other Types of Virtualization, Virtualization and Cloud Computing, Pros and Cons of Virtualization, Technology	**8 Hours**
Module – 2	
Cloud Computing Architecture, Introduction, Cloud Reference Model, Architecture, Infrastructure / Hardware as a Service, Platform as a Service, Software as a Service, Types of Clouds, Public Clouds, Private Clouds, Hybrid Clouds, Community Clouds, Economics of the Cloud, Open Challenges, Cloud Definition, Cloud Interoperability and Standards Scalability and Fault Tolerance Security, Trust, and Privacy Organizational Aspects Aneka: Cloud Application Platform, Framework Overview, Anatomy of the Aneka Container, From the Ground Up: Platform Abstraction Layer, Fabric Services, foundation Services, Application Services, Building Aneka Clouds, Infrastructure Organization, Logical Organization, Private Cloud Deployment Mode, Public Cloud Deployment Mode, Hybrid Cloud Deployment Mode, Cloud Programming and Management, Aneka SDK, Management Tools	**8 Hours**
Module – 3	
Concurrent Computing: Thread Programming, Introducing Parallelism for Single Machine Computation, Programming Applications with Threads, What is a Thread?, Thread APIs, Techniques for Parallel Computation with Threads, Multithreading with Aneka, Introducing the Thread Programming Model, Aneka Thread vs. Common Threads, Programming Applications with Aneka Threads, Aneka Threads Application Model, Domain Decomposition: Matrix Multiplication, Functional Decomposition: Sine, Cosine, and Tangent. High-Throughput Computing: Task Programming, Task Computing,	**8 Hours**

Characterizing a Task, Computing Categories, Frameworks for Task Computing, Task-based Application Models, Embarrassingly Parallel Applications, Parameter Sweep Applications, MPI Applications, Workflow Applications with Task Dependencies, Aneka Task-Based Programming, Task Programming Model, Developing Applications with the Task Model, Developing Parameter Sweep Application, Managing Workflows.	
Module – 4	
Data Intensive Computing: Map-Reduce Programming, What is Data-Intensive Computing?, Characterizing Data-Intensive Computations, Challenges Ahead, Historical Perspective, Technologies for Data-Intensive Computing, Storage Systems, Programming Platforms, Aneka MapReduce Programming, Introducing the MapReduce Programming Model, Example Application	**8 Hours**
Module – 5	
Cloud Platforms in Industry, Amazon Web Services, Compute Services, Storage Services, Communication Services, Additional Services, Google AppEngine, Architecture and Core Concepts, Application Life-Cycle, Cost Model, Observations, Microsoft Azure, Azure Core Concepts, SQL Azure, Windows Azure Platform Appliance. Cloud Applications Scientific Applications, Healthcare: ECG Analysis in the Cloud, , Social Networking, Media Applications, Multiplayer Online Gaming.	**8 Hours**
Course outcomes: The students should be able to:	
• Explain the concepts and terminologies of cloud computing • Demonstrate cloud frameworks and technologies • Define data intensive computing • Demonstrate cloud applications	
Question paper pattern: The question paper will have ten questions. There will be 2 questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer 5 full questions, selecting one full question from each module.	
Text Books: 1. Rajkumar Buyya, Christian Vecchiola, and Thamarai Selvi Mastering Cloud. Computing McGraw Hill Education	
Reference Books:	
NIL	

MOBILE APPLICATION DEVELOPMENT
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VI

Subject Code	15CS661	IA Marks	20
Number of Lecture Hours/Week	3	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 03

Course objectives: This course will enable students to

- Learn to setup Android application development environment
- Illustrate user interfaces for interacting with apps and triggering actions
- Interpret tasks used in handling multiple activities
- Identify options to save persistent application data
- Appraise the role of security and performance in Android applications

Module – 1	**Teaching Hours**
Get started, Build your first app, Activities, Testing, debugging and using support libraries	**8 Hours**
Module – 2	
User Interaction, Delightful user experience, Testing your UI	**8 Hours**
Module – 3	
Background Tasks, Triggering, scheduling and optimizing background tasks	**8 Hours**
Module – 4	
All about data, Preferences and Settings, Storing data using SQLite, Sharing data with content providers, Loading data using Loaders	**8 Hours**
Module – 5	
Permissions, Performance and Security, Firebase and AdMob, Publish	**8 Hours**

Course outcomes: The students should be able to:

- Create, test and debug Android application by setting up Android development environment
- Implement adaptive, responsive user interfaces that work across a wide range of devices.
- Infer long running tasks and background work in Android applications
- Demonstrate methods in storing, sharing and retrieving data in Android applications
- Analyze performance of android applications and understand the role of permissions and security
- Describe the steps involved in publishing Android application to share with the world

Question paper pattern:
The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

Text Books:

1. Google Developer Training, "Android Developer Fundamentals Course – Concept Reference", Google Developer Training Team, 2017. https://www.gitbook.com/book/google-developer-training/android-developer-fundamentals-course-concepts/details (Download pdf file from the above link)

Reference Books:

1. Erik Hellman, “Android Programming – Pushing the Limits”, 1st Edition, Wiley India Pvt Ltd, 2014.
2. Dawn Griffiths and David Griffiths, “Head First Android Development”, 1st Edition, O’Reilly SPD Publishers, 2015.
3. J F DiMarzio, “Beginning Android Programming with Android Studio”, 4th Edition, Wiley India Pvt Ltd, 2016. ISBN-13: 978-8126565580
4. Anubhav Pradhan, Anil V Deshpande, “ Composing Mobile Apps” using Android, Wiley 2014, ISBN: 978-81-265-4660-2

BIG DATA ANALYTICS
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VI

Subject Code	15CS662	IA Marks	20
Number of Lecture Hours/Week	4	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 03

Course objectives: This course will enable students to

- Interpret the data in the context of the business.
- Identify an appropriate method to analyze the data
- Show analytical model of a system

Module – 1	**Teaching Hours**
Introduction to Data Analytics and Decision Making: Introduction, Overview of the Book, The Methods, The Software, Modeling and Models, Graphical Models, Algebraic Models, Spreadsheet Models, Seven-Step Modeling Process.**Describing the Distribution of a Single Variable**:Introduction,Basic Concepts, Populations and Samples, Data Sets,Variables,and Observations, Types of Data, Descriptive Measures for Categorical Variables, Descriptive Measures for Numerical Variables, Numerical Summary Measures, Numerical Summary Measures with StatTools,Charts for Numerical Variables, Time Series Data, Outliers and Missing Values,Outliers,Missing Values, Excel Tables for Filtering,Sorting,and Summarizing. **Finding Relationships among Variables**: Introduction, Relationships among Categorical Variables, Relationships among Categorical Variables and a Numerical Variable, Stacked and Unstacked Formats, Relationships among Numerical Variables, Scatterplots, Correlation and Covariance, Pivot Tables.	**08 Hours**
Module – 2	
Probability and Probability Distributions:Introduction,Probability Essentials, Rule of Complements, Addition Rule, Conditional Probability and the Multiplication Rule, Probabilistic Independence, Equally Likely Events, Subjective Versus Objective Probabilities, Probability Distribution of a Single Random Variable, Summary Measures of a Probability Distribution, Conditional Mean and Variance, Introduction to Simulation. **Normal,Binormal,Poisson,and Exponential Distributions**:Introduction,The Normal Distribution, Continuous Distributions and Density Functions, The Normal Density,Standardizing:Z-Values,Normal Tables and Z-Values, Normal Calculations in Excel, Empirical Rules Revisited, Weighted Sums of Normal Random Variables, Applications of the Normal Random Distribution, The Binomial Distribution, Mean and Standard Deviation of the Binomial Distribution, The Binomial Distribution in the Context of Sampling, The Normal Approximation to the Binomial, Applications of the Binomial Distribution, The Poisson and Exponential Distributions, The Poisson Distribution, The Exponential Distribution.	**08 Hours**
Module – 3	
Decision Making under Uncertainty:Introduction,Elements of Decision Analysis, Payoff Tables, Possible Decision Criteria, Expected Monetary	**08 Hours**

Value(EMY),Sensitivity Analysis, Decision Trees, Risk Profiles, The Precision Tree Add-In,Bayes' Rule, Multistage Decision Problems and the Value of Information, The Value of Information, Risk Aversion and Expected Utility, Utility Functions, Exponential Utility, Certainty Equivalents, Is Expected Utility Maximization Used? **Sampling and Sampling Distributions**: Introduction, Sampling Terminology, Methods for Selecting Random Samples, Simple Random Sampling, Systematic Sampling, Stratified Sampling, Cluster Sampling, Multistage Sampling Schemes, Introduction to Estimation, Sources of Estimation Error, Key Terms in Sampling, Sampling Distribution of the Sample Mean, The Central Limit Theorem, Sample Size Selection, Summary of Key Ideas for Simple Random Sampling.	
Module – 4	
Confidence Interval Estimation: Introduction, Sampling Distributions, The t Distribution, Other Sampling Distributions, Confidence Interval for a Mean, Confidence Interval for a Total, Confidence Interval for a Proportion, Confidence Interval for a Standard Deviation, Confidence Interval for the Difference between Means, Independent Samples, Paired Samples, Confidence Interval for the Difference between Proportions, Sample Size Selection, Sample Size Selection for Estimation of the Mean, Sample Size Selection for Estimation of Other Parameters. **Hypothesis Testing**:Introduction,Concepts in Hypothesis Testing, Null and Alternative Hypothesis, One-Tailed Versus Two-Tailed Tests, Types of Errors, Significance Level and Rejection Region, Significance from p-values, Type II Errors and Power, Hypothesis Tests and Confidence Intervals, Practical versus Statistical Significance, Hypothesis Tests for a Population Mean, Hypothesis Tests for Other Parameters, Hypothesis Tests for a Population Proportion, Hypothesis Tests for Differences between Population Means, Hypothesis Test for Equal Population Variances, Hypothesis Tests for Difference between Population Proportions, Tests for Normality, Chi-Square Test for Independence.	**08 Hours**
Module – 5	
Regression Analysis: Estimating Relationships: Introduction, Scatterplots : Graphing Relationships, Linear versus Nonlinear Relationships,Outliers,Unequal Variance, No Relationship,Correlations:Indications of Linear Relationships, Simple Linear Regression, Least Squares Estimation, Standard Error of Estimate, The Percentage of Variation Explained:R-Square,Multiple Regression, Interpretation of Regression Coefficients, Interpretation of Standard Error of Estimate and R-Square, Modeling Possibilities, Dummy Variables, Interaction Variables, Nonlinear Transformations, Validation of the Fit. **Regression Analysis**: Statistical Inference:Introduction,The Statistical Model, Inferences About the Regression Coefficients, Sampling Distribution of the Regression Coefficients, Hypothesis Tests for the Regression Coefficients and p-Values, A Test for the Overall Fit: The ANOVA Table,Multicollinearity,Include/Exclude Decisions, Stepwise Regression,Outliers,Violations of Regression Assumptions,Nonconstant Error Variance,Nonnormality of Residuals,Autocorrelated Residuals ,Prediction.	**08 Hours**
Course outcomes: The students should be able to:	
• Explain the importance of data and data analysis • Interpret the probabilistic models for data • Define hypothesis, uncertainty principle	

- Evaluate regression analysis

Question paper pattern:

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. S C Albright and W L Winston, Business analytics: data analysis and decision making, 5/e Cenage Learning

Reference Books:

WIRELESS NETWORKS AND MOBILE COMPUTING
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VI

Subject Code	15CS663	IA Marks	20
Number of Lecture Hours/Week	3	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 03

Course objectives: This course will enable students to

- Describe the wireless communication.
- Illustrate operations involved in Mobile IP.
- Discover the concepts of mobile computing and databases.

Module – 1	**Teaching Hours**
Mobile Communication, Mobile Computing, Mobile Computing Architecture, Mobile Devices Mobile System Networks, Data Dissemination, Mobility Management, Security Cellular Networks and Frequency Reuse, Mobile Smartphone, Smart Mobiles, and Systems Handheld Pocket Computers, Handheld Devices, Smart Systems, Limitations of Mobile Devices Automotive Systems	**8 Hours**
Module – 2	
GSM-Services and System Architecture, Radio Interfaces of GSM, Protocols of GSM Localization, Call Handling Handover, Security, New Data Services, General Packet Radio Service High-speed Circuit Switched Data, DECT, Modulation, Multiplexing, Controlling the Medium Access Spread Spectrum, Frequency Hopping Spread Spectrum (FHSS),Coding Methods, Code Division Multiple Access, IMT-2000 3G Wireless Communication Standards, WCDMA 3G Communications Standards ,CDMMA2000 3G Communication Standards, I-mode, OFDM, High Speed Packet Access (HSPA) 3G Network Long-term Evolution, WiMax Rel 1.0 IEEE 802.16e, Broadband Wireless Access,4G Networks, Mobile Satellite Communication Networks	**8 Hours**
Module – 3	
IP and Mobile IP Network Layers, Packet Delivery and Handover Management Location Management, Registration, Tunnelling and Encapsulation, Route Optimization Dynamic Host Configuration Protocol, VoIP, IPsec Conventional TCP/IP Transport Layer Protocols, Indirect TCP, Snooping TCP Mobile TCP, Other Methods of Mobile TCP-layer Transmission ,TCP over 2.5G/3G Mobile Networks	**8 Hours**
Module – 4	
Data Organization, Database Transactional Models – ACID Rules, Query Processing Data Recovery Process, Database Hoarding Techniques , Data Caching, Client-Server Computing for Mobile Computing and Adaptation Adaptation Software for Mobile Computing, Power-Aware Mobile Computing, Context-aware Mobile Computing	**8 Hours**
Module – 5	
Communication Asymmetry, Classification of Data-delivery Mechanisms, Data Dissemination Broadcast Models, Selective Tuning and Indexing techniques, Digital Audio Broadcasting (DAB), Digital Video Broadcasting	**8 Hours**

Synchronization, Synchronization Software for Mobile Devices, Synchronization Software for Mobile Devices SyncML-Synchronization Language for Mobile Computing,Sync4J (Funambol), Synchronized Multimedia Markup Language (SMIL)	

Course outcomes: The students should be able to:

- Summarize various mobile communication systems.
- Describe various multiplexing systems used in mobile computing.
- Indicate the use and importance of data synchronization in mobile computing

Question paper pattern:

The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

Text Books:

1. Raj kamal: Mobile Computing, 2ND EDITION, Oxford University Press, 2007/2012
2. Martyn Mallik: Mobile and Wireless Design Essentials, Wiley India, 2003

Reference Books:

1. Ashok Talukder, Roopa Yavagal, Hasan Ahmed: Mobile Computing, Technology, Applications and Service Creation, 2nd Edition, Tata McGraw Hill, 2010.
2. Iti Saha Misra: Wireless Communications and Networks, 3G and Beyond, Tata McGraw Hill, 2009.

PYTHON APPLICATION PROGRAMMING
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VI

Subject Code	15CS664	IA Marks	20
Number of Lecture Hours/Week	3	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 03

Course objectives: This course will enable students to

- Learn Syntax and Semantics and create Functions in Python.
- Handle Strings and Files in Python.
- Understand Lists, Dictionaries and Regular expressions in Python.
- Implement Object Oriented Programming concepts in Python
- Build Web Services and introduction to Network and Database Programmingin Python.

Module – 1	**Teaching Hours**
Why should you learn to write programs, Variables, expressions and statements, Conditional execution, Functions	**8 Hours**
Module – 2	
Iteration, Strings, Files	**8 Hours**
Module – 3	
Lists, Dictionaries, Tuples, Regular Expressions	**8 Hours**
Module – 4	
Classes and objects, Classes and functions, Classes and methods	**8 Hours**
Module – 5	
Networked programs, Using Web Services, Using databases and SQL	**8 Hours**

Course outcomes: The students should be able to:

- Examine Python syntax and semantics and be fluent in the use of Python flow control and functions.
- Demonstrate proficiency in handling Strings and File Systems.
- Create, run and manipulate Python Programs using core data structures like Lists, Dictionaries and use Regular Expressions.
- Interpret the concepts of Object-Oriented Programming as used in Python.
- Implement exemplary applications related to Network Programming, Web Services and Databases in Python.

Question paper pattern:
The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

Text Books:

1. Charles R. Severance, "Python for Everybody: Exploring Data Using Python 3", 1st Edition, CreateSpace Independent Publishing Platform, 2016. (http://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf) (Chapters 1 – 13, 15)
2. Allen B. Downey, "Think Python: How to Think Like a Computer Scientist", 2nd Edition, Green Tea Press, 2015.

(http://greenteapress.com/thinkpython2/thinkpython2.pdf) (Chapters 15, 16, 17) (Download pdf files from the above links)

Reference Books:

1. Charles Dierbach, "Introduction to Computer Science Using Python", 1st Edition, Wiley India Pvt Ltd. ISBN-13: 978-8126556014
2. Mark Lutz, “Programming Python”, 4th Edition, O’Reilly Media, 2011.ISBN-13: 978-9350232873
3. Wesley J Chun, “Core Python Applications Programming”, 3rd Edition,Pearson Education India, 2015. ISBN-13: 978-9332555365
4. Roberto Tamassia, Michael H Goldwasser, Michael T Goodrich, “Data Structures and Algorithms in Python”,1stEdition, Wiley India Pvt Ltd, 2016. ISBN-13: 978-8126562176
5. Reema Thareja, “Python Programming using problem solving approach”, Oxford university press, 2017

SERVICE ORIENTED ARCHITECTURE
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VI

Subject Code	15CS665	IA Marks	20
Number of Lecture Hours/Week	3	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 03

Course objectives: This course will enable students to

- Compare various architecture for application development
- Illustrate the importance of SOA in Application Integration
- Learn web service and SOA related tools and governance

Module – 1	**Teaching Hours**
SOA BASICS: Software Architecture; Need for Software Architecture, Objectives of Software Architecture, Types of IT Architecture, Architecture Patterns and Styles, **Service oriented Architecture;** Service Orientation in Daily Life, Evolution of SOA, Drives for SOA, Dimension of SOA, Key components, perspective of SOA, **Enterprise-wide SOA;** Considerations for Enterprise -Wide SOA, Strawman Architecture For Enterprise-Wide-SOA-Enterprise, SOA-Layers, Application Development Process, SOA Methodology For Enterprise **Text 1: Ch2: 2.1 – 2.4; Ch3:3.1-3.7; Ch4: 4.1 – 4.5**	**8 Hours**
Module – 2	
Enterprise Applications; Architecture Considerations, Solution Architecture for enterprise application, **Software platforms for enterprise Applications;** Package Application Platforms, Enterprise Application Platforms, **Service-oriented-Enterprise Applications;** Considerations for Service-Oriented Enterprise Applications, Patterns for SOA, Pattern-Based Architecture for Service-Oriented Enterprise Application(java reference model only). Composite Applications, SOA programming models. **Text 1: Ch5:5.1, 5.2, 6.1, 6.2** (PageNo 74-81)**, 7.1 – 7.5**	**8 Hours**
Module – 3	
SOA ANALYSIS AND DESIGN; Need For Models, Principles of Service Design, Design of Activity Services, Design of Data sevices, Design of Client services and Design of business process services, **Technologies of SOA;** Technologies For Service Enablement, Technologies For Service Integration, Technologies for Service orchestration. **Text 1: Ch 8: 8.1 – 8.6, 9.1 – 9.3**	**8 Hours**
Module – 4	
Business case for SOA; Stakeholder OBJECTIVES, Benefits of SOA, Cost Savings, Return on Investment, SOA Governance, **Security and implementation;** SOA Governance, SOA Security, approach for enterprise wide SOA implementation, **Trends in SOA;** Technologies in Relation to SOA, Advances in SOA. **Text 1: Ch 10: 10.1 -10.4, Ch 11: 11.1 to 11.3, Ch12:12.2, 12.3**	**8 Hours**
Module – 5	
SOA Technologies-PoC; Loan Management System(LMS), PoC-Requirements Architectures of LMS **SOA based integration;** integrating existing application, **SOA best practices,** Basic SOA using REST. Role of WSDL,SOAP and	**8 Hours**

JAVA/XML Mapping in SOA. **Text 1:Page No 245-248; ReferenceBook:Chapter3; Text 1:Page No 307-310** **Text 2: Ch 3, Ch4**	

Course outcomes: The students should be able to:

- Compare the different IT architecture
- Analysis and design of SOA based applications
- Implementation of web service and realization of SOA
- Implementation of RESTful services

Question paper pattern:

The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

Text Books:

1. Shankar Kambhampaly, "Service–Oriented Architecture for Enterprise Applications",Wiley Second Edition, 2014.
2. Mark D. Hansen, "SOA using Java Web Services", Practice Hall, 2007.

Reference Books:

1. Waseem Roshen, "SOA-Based Enterprise Integration", Tata McGraw-HILL, 2009.

MULTI-CORE ARCHITECTURE AND PROGRAMMING
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VI

Subject Code	15CS666	IA Marks	20
Number of Lecture Hours/Week	3	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 03

Course objectives: This course will enable students to

- Explain the recent trends in the field of Computer Architecture and describe performance related parameters
- Illustrate the need for quasi-parallel processing.
- Formulate the problems related to multiprocessing
- Compare different types of multicore architectures

Module – 1	**Teaching Hours**
Introduction to Multi-core Architecture Motivation for Concurrency in software, Parallel Computing Platforms, Parallel Computing in Microprocessors, Differentiating Multi-core Architectures from Hyper- Threading Technology, Multi-threading on Single-Core versus Multi-Core Platforms Understanding Performance, Amdahl's Law, Growing Returns: Gustafson's Law. **System Overview of Threading** : Defining Threads, System View of Threads, Threading above the Operating System, Threads inside the OS, Threads inside the Hardware, What Happens When a Thread Is Created, Application Programming Models and Threading, Virtual Environment: VMs and Platforms, Runtime Virtualization, System Virtualization.	**8 Hours**
Module – 2	
Fundamental Concepts of Parallel Programming : Designing for Threads, Task Decomposition, Data Decomposition, Data Flow Decomposition, Implications of Different Decompositions, Challenges You'll Face, Parallel Programming Patterns, A Motivating Problem: Error Diffusion, Analysis of the Error Diffusion Algorithm, An Alternate Approach: Parallel Error Diffusion, Other Alternatives. **Threading and Parallel Programming Constructs:** Synchronization, Critical Sections, Deadlock, Synchronization Primitives, Semaphores, Locks, Condition Variables, Messages, Flow Control- based Concepts, Fence, Barrier, Implementation-dependent Threading Features	**8 Hours**
Module – 3	
Threading APIs : Threading APls for Microsoft Windows, Win32/MFC Thread APls, Threading APls for Microsoft. NET Framework, Creating Threads, Managing Threads, Thread Pools, Thread Synchronization, POSIX Threads, Creating Threads, Managing Threads, Thread Synchronization, Signaling, Compilation and Linking.	**8 Hours**
Module – 4	
OpenMP: A Portable Solution for Threading : Challenges in Threading a Loop, Loop-carried Dependence, Data-race Conditions, Managing Shared and Private Data, Loop Scheduling and Portioning, Effective Use of Reductions, Minimizing Threading Overhead, Work-sharing Sections, Performance-oriented Programming, Using Barrier and No wait, Interleaving Single-thread and Multi-thread Execution, Data Copy-in and Copy-out, Protecting Updates of Shared	**8 Hours**

Variables, Intel Task queuing Extension to OpenMP, OpenMP Library Functions, OpenMP Environment Variables, Compilation, Debugging, performance	
Module – 5	
Solutions to Common Parallel Programming Problems : Too Many Threads, Data Races, Deadlocks, and Live Locks, Deadlock, Heavily Contended Locks, Priority Inversion, Solutions for Heavily Contended Locks, Non-blocking Algorithms, ABA Problem, Cache Line Ping-ponging, Memory Reclamation Problem, Recommendations, Thread-safe Functions and Libraries, Memory Issues, Bandwidth, Working in the Cache, Memory Contention, Cache-related Issues, False Sharing, Memory Consistency, Current IA-32 Architecture, Itanium Architecture, High-level Languages, Avoiding Pipeline Stalls on IA-32,Data Organization for High Performance.	**8 Hours**
Course outcomes: The students should be able to:	
• Identify the issues involved in multicore architectures • Explain fundamental concepts of parallel programming and its design issues • Solve the issues related to multiprocessing and suggest solutions • Point out the salient features of different multicore architectures and how they exploit parallelism • Illustrate OpenMP and programming concept	
Question paper pattern: The question paper will have TEN questions. There will be TWO questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer FIVE full questions, selecting ONE full question from each module.	
Text Books:	
1. Multicore Programming , Increased Performance through Software Multi-threading by Shameem Akhter and Jason Roberts , Intel Press , 2006	
Reference Books:	
NIL	

MANAGEMENT AND ENTREPRENEURSHIP FOR IT INDUSTRY
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – V

Subject Code	15CS51	IA Marks	20
Number of Lecture Hours/Week	4	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03

CREDITS – 04

Course objectives: This course will enable students to

- Explain the principles of management, organization and entrepreneur.
- Discuss on planning, staffing, ERP and their importance
- Infer the importance of intellectual property rights and relate the institutional support

Module – 1	**Teaching Hours**
Introduction – Meaning, nature and characteristics of management, scope and functional areas of management, goals of management, levels of management, brief overview of evolution of management. Planning- Nature, importance, types of plans, steps in planning, Organizing- nature and purpose, types of organization.	**10 Hours**
Module – 2	
Staffing- meaning, process of recruitment and selection. Directing and controlling- meaning and nature of directing, leadership styles, motivation theories. Controlling- meaning, steps in controlling, methods of establishing control, Communication- Meaning and importance, Coordination- meaning and importance	**10 Hours**
Module – 3	
Entrepreneur – meaning of entrepreneur, types of entrepreneurship, stages of entrepreneurial process, role of entrepreneurs in economic development, entrepreneurship in India, barriers to entrepreneurship. Identification of business opportunities- market feasibility study, technical feasibility study, financial feasibility study and social feasibility study.	**10 Hours**
Module – 4	
Preparation of project and ERP - meaning of project, project identification, project selection, project report, need and significance of report, contents, formulation, guidelines by planning commission for project report **Enterprise Resource Planning: Meaning and Importance- ERP** and Functional areas of Management – Marketing / Sales- Supply Chain Management – Finance and Accounting – Human Resources – Types of reports and methods of report generation	**10 Hours**
Module – 5	
Micro and Small Enterprises: Definition of micro and small enterprises, characteristics and advantages of micro and small enterprises, steps in establishing micro and small enterprises, Government of India indusial policy 2007 on micro and small enterprises, case study (Microsoft), Case study(Captain G R Gopinath),case study (N R Narayana Murthy & Infosys), **Institutional support:** MSME-DI, NSIC, SIDBI, KIADB, KSSIDC, TECSOK, KSFC, DIC and District level single window agency, **Introduction to IPR.**	**10 Hours**

Course outcomes: The students should be able to:

- Define management, organization, entrepreneur, planning, staffing, ERP and outline

their importance in entrepreneurship
- Utilize the resources available effectively through ERP
- Make us of IPRs and institutional support in entrepreneurship

Question paper pattern:

The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

Text Books:

1. Principles of Management -P. C. Tripathi, P. N. Reddy; Tata McGraw Hill, 4th / 6th Edition, 2010.
2. Dynamics of Entrepreneurial Development & Management -Vasant Desai Himalaya Publishing House.
3. Entrepreneurship Development -Small Business Enterprises -Poornima M Charantimath Pearson Education – 2006.
4. Management and Enterpreneurship- Kanishka Bedi- Oxford University Press-2017

Reference Books:

1. Management Fundamentals -Concepts, Application, Skill Development Robert Lusier – Thomson.
2. Entrepreneurship Development -S S Khanka -S Chand & Co.
3. Management -Stephen Robbins -Pearson Education /PHI -17th Edition, 2003

COMPUTER NETWORKS
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – V

Subject Code	15CS52	IA Marks	20
Number of Lecture Hours/Week	4	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03

CREDITS – 04

Course objectives: This course will enable students to

- Demonstration of application layer protocols
- Discuss transport layer services and understand UDP and TCP protocols
- Explain routers, IP and Routing Algorithms in network layer
- Disseminate the Wireless and Mobile Networks covering IEEE 802.11 Standard
- Illustrate concepts of Multimedia Networking, Security and Network Management

Module – 1	**Teaching Hours**
Application Layer: Principles of Network Applications: Network Application Architectures, Processes Communicating, Transport Services Available to Applications, Transport Services Provided by the Internet, Application-Layer Protocols. The Web and HTTP: Overview of HTTP, Non-persistent and Persistent Connections, HTTP Message Format, User-Server Interaction: Cookies, Web Caching, The Conditional GET, File Transfer: FTP Commands & Replies, Electronic Mail in the Internet: SMTP, Comparison with HTTP, Mail Message Format, Mail Access Protocols, DNS; The Internet's Directory Service: Services Provided by DNS, Overview of How DNS Works, DNS Records and Messages, Peer-to-Peer Applications: P2P File Distribution, Distributed Hash Tables, Socket Programming: creating Network Applications: Socket Programming with UDP, Socket Programming with TCP. **T1: Chap 2**	**10 Hours**
Module – 2	
Transport Layer : Introduction and Transport-Layer Services: Relationship Between Transport and Network Layers, Overview of the Transport Layer in the Internet, Multiplexing and Demultiplexing: Connectionless Transport: UDP,UDP Segment Structure, UDP Checksum, Principles of Reliable Data Transfer: Building a Reliable Data Transfer Protocol, Pipelined Reliable Data Transfer Protocols, Go-Back-N, Selective repeat, Connection-Oriented Transport TCP: The TCP Connection, TCP Segment Structure, Round-Trip Time Estimation and Timeout, Reliable Data Transfer, Flow Control, TCP Connection Management, Principles of Congestion Control: The Causes and the Costs of Congestion, Approaches to Congestion Control, Network-assisted congestion-control example, ATM ABR Congestion control, TCP Congestion Control: Fairness. **T1: Chap 3**	**10 Hours**
Module – 3	
The Network layer: What's Inside a Router?: Input Processing, Switching, Output Processing, Where Does Queuing Occur? Routing control plane, IPv6,A Brief foray into IP Security, Routing Algorithms: The Link-State (LS) Routing Algorithm, The Distance-Vector (DV) Routing Algorithm, Hierarchical Routing,	**10 Hours**

Routing in the Internet, Intra-AS Routing in the Internet: RIP, Intra-AS Routing in the Internet: OSPF, Inter/AS Routing: BGP, Broadcast and Multicast Routing: Broadcast Routing Algorithms and Multicast. **T1: Chap 4:4.3-4.7**	
Module – 4	
Wireless and Mobile Networks: Cellular Internet Access: An Overview of Cellular Network Architecture, 3G Cellular Data Networks: Extending the Internet to Cellular subscribers, On to 4G:LTE,Mobility management: Principles, Addressing, Routing to a mobile node, Mobile IP, Managing mobility in cellular Networks, Routing calls to a Mobile user, Handoffs in GSM, Wireless and Mobility: Impact on Higher-layer protocols. **T1: Chap: 6 : 6.4-6.8**	**10 Hours**
Module – 5	
Multimedia Networking: Properties of video, properties of Audio, Types of multimedia Network Applications, Streaming stored video: UDP Streaming, HTTP Streaming, Adaptive streaming and DASH, content distribution Networks, case studies: Netflix, You Tube and Kankan. **Network Support for Multimedia:** Dimensioning Best-Effort Networks, Providing Multiple Classes of Service, Diffserv, Per-Connection Quality-of-Service (QoS) Guarantees: Resource Reservation and Call Admission **T1: Chap: 7: 7.1,7.2,7.5**	**10 Hours**

Course outcomes: The students should be able to:

- Explain principles of application layer protocols
- Recognize transport layer services and infer UDP and TCP protocols
- Classify routers, IP and Routing Algorithms in network layer
- Understand the Wireless and Mobile Networks covering IEEE 802.11 Standard
- Describe Multimedia Networking and Network Management

Question paper pattern:

The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

Text Books:

1. James F Kurose and Keith W Ross, Computer Networking, A Top-Down Approach, Sixth edition, Pearson,2017 .

Reference Books:

1. Behrouz A Forouzan, Data and Communications and Networking, Fifth Edition, McGraw Hill, Indian Edition
2. Larry L Peterson and Brusce S Davie, Computer Networks, fifth edition, ELSEVIER
3. Andrew S Tanenbaum, Computer Networks, fifth edition, Pearson
4. Mayank Dave, Computer Networks, Second edition, Cengage Learning

DATABASE MANAGEMENT SYSTEM
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – V

Subject Code	15CS53	IA Marks	20
Number of Lecture Hours/Week	4	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03

CREDITS – 04

Course objectives: This course will enable students to

- Provide a strong foundation in database concepts, technology, and practice.
- Practice SQL programming through a variety of database problems.
- Demonstrate the use of concurrency and transactions in database
- Design and build database applications for real world problems.

Module – 1	**Teaching Hours**
Introduction to Databases: Introduction, Characteristics of database approach, Advantages of using the DBMS approach, History of database applications. **Overview of Database Languages and Architectures:** Data Models, Schemas, and Instances. Three schema architecture and data independence, database languages, and interfaces, The Database System environment. **Conceptual Data Modelling using Entities and Relationships:** Entity types, Entity sets, attributes, roles, and structural constraints, Weak entity types, ER diagrams, examples, Specialization and Generalization. **Textbook 1:Ch 1.1 to 1.8, 2.1 to 2.6, 3.1 to 3.10**	**10 Hours**
Module – 2	
Relational Model: Relational Model Concepts, Relational Model Constraints and relational database schemas, Update operations, transactions, and dealing with constraint violations. **Relational Algebra:** Unary and Binary relational operations, additional relational operations (aggregate, grouping, etc.) Examples of Queries in relational algebra. **Mapping Conceptual Design into a Logical Design:** Relational Database Design using ER-to-Relational mapping. **SQL:** SQL data definition and data types, specifying constraints in SQL, retrieval queries in SQL, INSERT, DELETE, and UPDATE statements in SQL, Additional features of SQL. **Textbook 1: Ch4.1 to 4.5, 5.1 to 5.3, 6.1 to 6.5, 8.1; Textbook 2: 3.5**	**10 Hours**
Module – 3	
SQL : Advances Queries: More complex SQL retrieval queries, Specifying constraints as assertions and action triggers, Views in SQL, Schema change statements in SQL. **Database Application Development:** Accessing databases from applications, An introduction to JDBC, JDBC classes and interfaces, SQLJ, Stored procedures, Case study: The internet Bookshop. **Internet Applications:** The three-Tier application architecture, The presentation layer, The Middle Tier **Textbook 1: Ch7.1 to 7.4; Textbook 2: 6.1 to 6.6, 7.5 to 7.7.**	**10 Hours**
Module – 4	
Normalization: Database Design Theory – Introduction to Normalization using Functional and Multivalued Dependencies: Informal design guidelines for relation schema, Functional Dependencies, Normal Forms based on Primary Keys, Second and Third Normal Forms, Boyce-Codd Normal Form, Multivalued Dependency and Fourth Normal Form, Join Dependencies and Fifth Normal	**10 Hours**

Form. **Normalization Algorithms:** Inference Rules, Equivalence, and Minimal Cover, Properties of Relational Decompositions, Algorithms for Relational Database Schema Design, Nulls, Dangling tuples, and alternate Relational Designs, Further discussion of Multivalued dependencies and 4NF, Other dependencies and Normal Forms **Textbook 1: Ch14.1 to 14.7, 15.1 to 15.6**	
Module – 5	
Transaction Processing: Introduction to Transaction Processing, Transaction and System concepts, Desirable properties of Transactions, Characterizing schedules based on recoverability, Characterizing schedules based on Serializability, Transaction support in SQL. **Concurrency Control in Databases:** Two-phase locking techniques for Concurrency control, Concurrency control based on Timestamp ordering, Multiversion Concurrency control techniques, Validation Concurrency control techniques, Granularity of Data items and Multiple Granularity Locking. **Introduction to Database Recovery Protocols:** Recovery Concepts, NO-UNDO/REDO recovery based on Deferred update, Recovery techniques based on immediate update, Shadow paging, Database backup and recovery from catastrophic failures **Textbook 1: 20.1 to 20.6, 21.1 to 21.7, 22.1 to 22.4, 22.7.**	**10 Hours**
Course outcomes: The students should be able to:	
• Identify, analyze and define database objects, enforce integrity constraints on a database using RDBMS. • Use Structured Query Language (SQL) for database manipulation. • Design and build simple database systems • Develop application to interact with databases.	
Question paper pattern: The question paper will have TEN questions. There will be TWO questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer FIVE full questions, selecting ONE full question from each module.	
Text Books:	
1. Fundamentals of Database Systems, RamezElmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson. 2. Database management systems, Ramakrishnan, and Gehrke, 3rd Edition, 2014, McGraw Hill	
Reference Books:	
1. Silberschatz Korth and Sudharshan, Database System Concepts, 6th Edition, Mc-GrawHill, 2013. 2. Coronel, Morris, and Rob, Database Principles Fundamentals of Design, Implementation and Management, Cengage Learning 2012.	

AUTOMATA THEORY AND COMPUTABILITY
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – V

Subject Code	15CS54	IA Marks	20
Number of Lecture Hours/Week	4	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03

CREDITS – 04

Course objectives: This course will enable students to

- Introduce core concepts in Automata and Theory of Computation
- Identify different Formal language Classes and their Relationships
- Design Grammars and Recognizers for different formal languages
- Prove or disprove theorems in automata theory using their properties
- Determine the decidability and intractability of Computational problems

Module – 1	**Teaching Hours**
Why study the Theory of Computation, Languages and Strings: Strings, Languages. A Language Hierarchy, Computation, **Finite State Machines (FSM)**: Deterministic FSM, Regular languages, Designing FSM, Nondeterministic FSMs, From FSMs to Operational Systems, Simulators for FSMs, Minimizing FSMs, Canonical form of Regular languages, Finite State Transducers, Bidirectional Transducers. **Textbook 1: Ch 1,2, 3,4, 5.1 to 5.10**	**10 Hours**
Module – 2	
Regular Expressions (RE): what is a RE?, Kleene's theorem, Applications of REs, Manipulating and Simplifying REs. Regular Grammars: Definition, Regular Grammars and Regular languages. Regular Languages (RL) and Non-regular Languages: How many RLs, To show that a language is regular, Closure properties of RLs, to show some languages are not RLs. **Textbook 1: Ch 6, 7, 8: 6.1 to 6.4, 7.1, 7.2, 8.1 to 8.4**	**10 Hours**
Module – 3	
Context-Free Grammars(CFG): Introduction to Rewrite Systems and Grammars, CFGs and languages, designing CFGs, simplifying CFGs, proving that a Grammar is correct, Derivation and Parse trees, Ambiguity, Normal Forms. Pushdown Automata (PDA): Definition of non-deterministic PDA, Deterministic and Non-deterministic PDAs, Non-determinism and Halting, alternative equivalent definitions of a PDA, alternatives that are not equivalent to PDA. **Textbook 1: Ch 11, 12: 11.1 to 11.8, 12.1, 12.2, 12,4, 12.5, 12.6**	**10 Hours**
Module – 4	
Context-Free and Non-Context-Free Languages: Where do the Context-Free Languages(CFL) fit, Showing a language is context-free, Pumping theorem for CFL, Important closure properties of CFLs, Deterministic CFLs. Algorithms and Decision Procedures for CFLs: Decidable questions, Un-decidable questions. Turing Machine: Turing machine model, Representation, Language acceptability by TM, design of TM, Techniques for TM construction. **Textbook 1: Ch 13: 13.1 to 13.5, Ch 14: 14.1, 14.2, Textbook 2: Ch 9.1 to 9.6**	**10 Hours**
Module – 5	
Variants of Turing Machines (TM), The model of Linear Bounded automata: Decidability: Definition of an algorithm, decidability, decidable languages,	**10 Hours**

Undecidable languages, halting problem of TM, Post correspondence problem. Complexity: Growth rate of functions, the classes of P and NP, Quantum Computation: quantum computers, Church-Turing thesis. **Textbook 2: Ch 9.7 to 9.8, 10.1 to 10.7, 12.1, 12.2, 12.8, 12.8.1, 12.8.2**	

Course outcomes: The students should be able to:

- Acquire fundamental understanding of the core concepts in automata theory and Theory of Computation
- Learn how to translate between different models of Computation (e.g., Deterministic and Non-deterministic and Software models).
- Design Grammars and Automata (recognizers) for different language classes and become knowledgeable about restricted models of Computation (Regular, Context Free) and their relative powers.
- Develop skills in formal reasoning and reduction of a problem to a formal model, with an emphasis on semantic precision and conciseness.
- Classify a problem with respect to different models of Computation.

Question paper pattern:

The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

Text Books:

1. Elaine Rich, Automata, Computability and Complexity, 1st Edition, Pearson Education,2012/2013
2. K L P Mishra, N Chandrasekaran , 3rd Edition, Theory of Computer Science, PhI, 2012.

Reference Books:

1. John E Hopcroft, Rajeev Motwani, Jeffery D Ullman, Introduction to AutomataTheory, Languages, and Computation, 3rd Edition, Pearson Education, 2013
2. Michael Sipser : Introduction to the Theory of Computation, 3rd edition, Cengage learning,2013
3. John C Martin, Introduction to Languages and The Theory of Computation, 3rd Edition, Tata McGraw –Hill Publishing Company Limited, 2013
4. Peter Linz, “An Introduction to Formal Languages and Automata”, 3rd Edition, Narosa Publishers, 1998
5. Basavaraj S. Anami, Karibasappa K G, Formal Languages and Automata theory, Wiley India, 2012
6. C K Nagpal, Formal Languages and Automata Theory, Oxford University press, 2012.

OBJECT ORIENTED MODELING AND DESIGN
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – V

Subject Code	15CS551	IA Marks	20
Number of Lecture Hours/Week	3	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 03

Course objectives: This course will enable students to

- Describe the concepts involved in Object-Oriented modelling and their benefits.
- Demonstrate concept of use-case model, sequence model and state chart model for a given problem.
- Explain the facets of the unified process approach to design and build a Software system.
- Translate the requirements into implementation for Object Oriented design.
- Choose an appropriate design pattern to facilitate development procedure.

Module – 1	**Teaching Hours**
Introduction, Modelling Concepts and Class Modelling: What is Object orientation? What is OO development? OO Themes; Evidence for usefulness of OO development; OO modelling history. Modelling as Design technique: Modelling; abstraction; The Three models. Class Modelling: Object and Class Concept; Link and associations concepts; Generalization and Inheritance; A sample class model; Navigation of class models; Advanced Class Modelling, Advanced object and class concepts; Association ends; N-ary associations; Aggregation; Abstract classes; Multiple inheritance; Metadata; Reification; Constraints; Derived Data; Packages. **Text Book-1: Ch 1, 2, 3 and 4**	**8 Hours**
Module – 2	
UseCase Modelling and Detailed Requirements: Overview; Detailed object-oriented Requirements definitions; System Processes-A use case/Scenario view; Identifying Input and outputs-The System sequence diagram; Identifying Object Behaviour-The state chart Diagram; Integrated Object-oriented Models. **Text Book-2:Chapter- 6:Page 210 to 250**	**8 Hours**
Module – 3	
Process Overview, System Conception and Domain Analysis: Process Overview: Development stages; Development life Cycle; System Conception: Devising a system concept; elaborating a concept; preparing a problem statement. Domain Analysis: Overview of analysis; Domain Class model: Domain state model; Domain interaction model; Iterating the analysis. **Text Book-1:Chapter- 10,11,and 12**	**8 Hours**
Module – 4	
Use case Realization :The Design Discipline within up iterations: Object Oriented Design-The Bridge between Requirements and Implementation; Design Classes and Design within Class Diagrams; Interaction Diagrams-Realizing Use Case and defining methods; Designing with Communication Diagrams; Updating the Design Class Diagram; Package Diagrams-Structuring the Major Components; Implementation Issues for Three-Layer Design. **Text Book-2: Chapter 8: page 292 to 346**	**8 Hours**

Module – 5	
Design Patterns: Introduction; what is a design pattern?, Describing design patterns, the catalog of design patterns, Organizing the catalog, How design patterns solve design problems, how to select a design patterns, how to use a design pattern; Creational patterns: prototype and singleton(only);structural patterns adaptor and proxy(only). **Text Book-3:Chapter-1: 1.1, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8,Chapter-3,Chapter-4.**	**8 Hours**
Course outcomes: The students should be able to: • Describe the concepts of object-oriented and basic class modelling. • Draw class diagrams, sequence diagrams and interaction diagrams to solve problems. • Choose and apply a befitting design pattern for the given problem.	
Question paper pattern: The question paper will have TEN questions. There will be TWO questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer FIVE full questions, selecting ONE full question from each module.	
Text Books:	
1. Michael Blaha, James Rumbaugh: Object Oriented Modelling and Design with UML,2nd Edition, Pearson Education,2005 2. Satzinger, Jackson and Burd: Object-Oriented Analysis & Design with the Unified Process, Cengage Learning,2005. 3. Erich Gamma, Richard Helm, Ralph Johnson and john Vlissides: Design Patterns – Elements of Reusable Object-Oriented Software, Pearson Education,2007.	
Reference Books:	
1. Grady Booch et.al.: Object-Oriented Analysis and Design with Applications,3rd Edition,Pearson Education,2007. 2. 2.Frank Buschmann, RegineMeunier, Hans Rohnert, Peter Sommerlad, Michel Stal: Pattern –Oriented Software Architecture. A system of Patterns , Volume 1, John Wiley and Sons.2007. 3. 3. Booch, Jacobson, Rambaugh : Object-Oriented Analysis and Design with Applications, 3rd edition, pearson, Reprint 2013	

SOCIAL NETWORK ANALYSIS
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – V

Subject Code	15IS552	IA Marks	20
Number of Lecture Hours/Week	03	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 03

Course objectives: This course will enable students to

- Discuss essential knowledge of network analysis applicable to real world data, with examples from today's most popular social networks.

Module 1	**Teaching Hours**
Introduction to social network analysis and Descriptive network analysis: Introduction to new science of networks. Networks examples. Graph theory basics. Statistical network properties. Degree distribution, clustering coefficient. Frequent patterns. Network motifs. Cliques and k-cores.	**8 Hours**
Module 2	
Network structure, Node centralities and ranking on network: Nodes and edges, network diameter and average path length. Node centrality metrics: degree, closeness and betweenness centrality. Eigenvector centrality and PageRank. Algorithm HITS.	**8 Hours**
Module 3	
Network communities and Affiliation networks: Networks communities. Graph partitioning and cut metrics. Edge betweenness. Modularity clustering. Affiliation network and bipartite graphs. 1-mode projections. Recommendation systems.	**8 Hours**
Module 4	
Information and influence propagation on networks and Network visualization: Social Diffusion. Basic cascade model. Influence maximization. Most influential nodes in network. Network visualization and graph layouts. Graph sampling. Low -dimensional projections	**8 Hours**
Module 5	
Social media mining and SNA in real world: FB/VK and Twitter analysis: Natural language processing and sentiment mining. Properties of large social networks: friends, connections, likes, re-tweets.	**8 Hours**

Course Outcomes: The students should be able to:

- Define notation and terminology used in network science.
- Demonstrate, summarize and compare networks.
- Explain basic principles behind network analysis algorithms.
- Analyzing real world network.

Question paper pattern:
The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

Text Books:

1. David Easley and John Kleinberg. "Networks, Crowds, and Markets: Reasoning About a Highly Connected World." Cambridge University Press 2010.

2. Eric Kolaczyk, Gabor Csardi. “Statistical Analysis of Network Data with R (Use R!)”. Springer, 2014.
3. Stanley Wasserman and Katherine Faust. "Social Network Analysis. Methods and Applications." Cambridge University Press, 1994.

Reference Books:

1. **NIL**

ADVANCED JAVA AND J2EE
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – V

Subject Code	15CS553	IA Marks	20
Number of Lecture Hours/Week	3	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 03

Course objectives: This course will enable students to

- Identify the need for advanced Java concepts like Enumerations and Collections
- Construct client-server applications using Java socket API
- Make use of JDBC to access database through Java Programs
- Adapt servlets to build server side programs
- Demonstrate the use of JavaBeans to develop component-based Java software

Module – 1	**Teaching Hours**
Enumerations, Autoboxing and Annotations(metadata): Enumerations, Enumeration fundamentals, the values() and valueOf() Methods, java enumerations are class types, enumerations Inherits Enum, example, type wrappers, Autoboxing, Autoboxing and Methods, Autoboxing/Unboxing occurs in Expressions, Autoboxing/Unboxing, Boolean and character values, Autoboxing/Unboxing helps prevent errors, A word of Warning. Annotations, Annotation basics, specifying retention policy, Obtaining Annotations at run time by use of reflection, Annotated element Interface, Using Default values, Marker Annotations, Single Member annotations, Built-In annotations.	**8 Hours**
Module – 2	
The collections and Framework: Collections Overview, Recent Changes to Collections, The Collection Interfaces, The Collection Classes, Accessing a collection Via an Iterator, Storing User Defined Classes in Collections, The Random Access Interface, Working With Maps, Comparators, The Collection Algorithms, Why Generic Collections?, The legacy Classes and Interfaces, Parting Thoughts on Collections.	**8 Hours**
Module – 3	
String Handling **:**The String Constructors, String Length, Special String Operations, String Literals, String Concatenation, String Concatenation with Other Data Types, String Conversion and toString( ) Character Extraction, charAt( ), getChars( ), getBytes( ) toCharArray(), String Comparison, equals( ) and equalsIgnoreCase( ), regionMatches( ) startsWith( ) and endsWith( ), equals( ) Versus == , compareTo( ) Searching Strings, Modifying a String, substring( ), concat( ), replace( ), trim( ), Data Conversion Using valueOf( ), Changing the Case of Characters Within a String, Additional String Methods, StringBuffer , StringBuffer Constructors, length( ) and capacity( ), ensureCapacity( ), setLength( ), charAt( ) and setCharAt( ), getChars( ),append( ), insert( ), reverse( ), delete( ) and deleteCharAt( ), replace( ), substring( ), Additional StringBuffer Methods, StringBuilder **Text Book 1: Ch 15**	**8 Hours**
Module – 4	

Background; The Life Cycle of a Servlet; Using Tomcat for Servlet Development; A simple Servlet; The Servlet API; The Javax.servlet Package; Reading Servlet Parameter; The Javax.servlet.http package; Handling HTTP Requests and Responses; Using Cookies; Session Tracking. Java Server Pages (JSP): JSP, JSP Tags, Tomcat, Request String, User Sessions, Cookies, Session Objects **Text Book 1: Ch 31 Text Book 2: Ch 11**	**8 Hours**
Module – 5	
The Concept of JDBC; JDBC Driver Types; JDBC Packages; A Brief Overview of the JDBC process; Database Connection; Associating the JDBC/ODBC Bridge with the Database; Statement Objects; ResultSet; Transaction Processing; Metadata, Data types; Exceptions. **Text Book 2: Ch 06**	**8 Hours**

Course outcomes: The students should be able to:

- Interpret the need for advanced Java concepts like enumerations and collections in developing modular and efficient programs
- Build client-server applications and TCP/IP socket programs
- Illustrate database access and details for managing information using the JDBC API
- Describe how servlets fit into Java-based web application architecture
- Develop reusable software components using Java Beans

Question paper pattern:

The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

Text Books:

1. Herbert Schildt: JAVA the Complete Reference, 7th/9th Edition, Tata McGraw Hill, 2007.
2. Jim Keogh: J2EE-TheCompleteReference, McGraw Hill, 2007.

Reference Books:

1. Y. Daniel Liang: Introduction to JAVA Programming, 7thEdition, Pearson Education, 2007.
2. Stephanie Bodoff et al: The J2EE Tutorial, 2nd Edition, Pearson Education,2004.
3. Uttam K Roy, Advanced JAVA programming, Oxford University press, 2015.

PROGRAMMING LANGAUGES
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – V

Subject Code	15IS554	IA Marks	20
Number of Lecture Hours/Week	3	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 03

Course objectives: This course will enable students to

- Acquaint with discipline of programming
- Familiarize with semantics of different constructs of languages
- Introduce different programming paradigms
- Illustrate use of different languages and their applications

Module – 1	**Teaching Hours**
Overview, Names, Types, Type systems	**8 Hours**
Module – 2	
Semantics, semantic interpretation	**8 Hours**
Module – 3	
Functions, function implementation, memory management	**8 Hours**
Module – 4	
Imperative programming, object oriented programming, functional programming	**8 Hours**
Module – 5	
Logic programming, event-driven programming, concurrent programming	**8 Hours**

Course outcomes: The students should be able to:

- Select appropriate languages for given applications
- Demonstrate usage and justification of different languages
- Compare and contrast the strengths and weaknesses of different languages

Question paper pattern:
The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

Text Books:

1. Programming languages by Allen B. Tucker and Robert E. Noonan

Reference Books:

NIL

COMPUTER NETWORK LABORATORY
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – V

Subject Code	15CSL57	IA Marks	20
Number of Lecture Hours/Week	01I + 02P	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 02

Course objectives: This course will enable students to

- Demonstrate operation of network and its management commands
- Simulate and demonstrate the performance of GSM and CDMA
- Implement data link layer and transport layer protocols.

Description (If any):

For the experiments below modify the topology and parameters set for the experiment and take multiple rounds of reading and analyze the results available in log files. Plot necessary graphs and conclude. Use NS2/NS3.

Lab Experiments:

PART A

1. Implement three nodes point – to – point network with duplex links between them. Set the queue size, vary the bandwidth and find the number of packets dropped.
2. Implement transmission of ping messages/trace route over a network topology consisting of 6 nodes and find the number of packets dropped due to congestion.
3. Implement an Ethernet LAN using n nodes and set multiple traffic nodes and plot congestion window for different source / destination.
4. Implement simple ESS and with transmitting nodes in wire-less LAN by simulation and determine the performance with respect to transmission of packets.
5. Implement and study the performance of GSM on NS2/NS3 (Using MAC layer) or equivalent environment.
6. Implement and study the performance of CDMA on NS2/NS3 (Using stack called Call net) or equivalent environment.

PART B

Implement the following in Java:

7. Write a program for error detecting code using CRC-CCITT (16- bits).
8. Write a program to find the shortest path between vertices using bellman-ford algorithm.
9. Using TCP/IP sockets, write a client – server program to make the client send the file name and to make the server send back the contents of the requested file if present.
10. Write a program on datagram socket for client/server to display the messages on client side, typed at the server side.
11. Write a program for simple RSA algorithm to encrypt and decrypt the data.
12. Write a program for congestion control using leaky bucket algorithm.

Study Experiment / Project:

NIL

Course outcomes: The students should be able to:

- Analyze and Compare various networking protocols.
- Demonstrate the working of different concepts of networking.

- Implement, analyze and evaluate networking protocols in NS2 / NS3

Conduction of Practical Examination:

1. All laboratory experiments are to be included for practical examination.
2. Students are allowed to pick one experiment from part A and part B with lot.
3. Strictly follow the instructions as printed on the cover page of answer script
4. Marks distribution: Procedure + Conduction + Viva: 80

   Part A: 10+25+5 =40

   Part B: 10+25+5 =40

5. Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.

DBMS LABORATORY WITH MINI PROJECT
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – V

Subject Code	15CSL58	IA Marks	20
Number of Lecture Hours/Week	01I + 02P	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 02

Course objectives: This course will enable students to

- Foundation knowledge in database concepts, technology and practice to groom students into well-informed database application developers.
- Strong practice in SQL programming through a variety of database problems.
- Develop database applications using front-end tools and back-end DBMS.

Description (If any):

PART-A: SQL Programming (Max. Exam Mks. 50)

- Design, develop, and implement the specified queries for the following problems using Oracle, MySQL, MS SQL Server, or any other DBMS under LINUX/Windows environment.
- Create Schema and insert at least 5 records for each table. Add appropriate database constraints.

PART-B: Mini Project (Max. Exam Mks. 30)

- Use Java, C#, PHP, Python, or any other similar front-end tool. All applications must be demonstrated on desktop/laptop as a stand-alone or web based application (Mobile apps on Android/IOS are not permitted.)

Lab Experiments:

Part A: SQL Programming

1 Consider the following schema for a Library Database:
BOOK(Book_id, Title, Publisher_Name, Pub_Year)
BOOK_AUTHORS(Book_id, Author_Name)
PUBLISHER(Name, Address, Phone)
BOOK_COPIES(Book_id, Branch_id, No-of_Copies)
BOOK_LENDING(Book_id, Branch_id, Card_No, Date_Out, Due_Date)
LIBRARY_BRANCH(Branch_id, Branch_Name, Address)
Write SQL queries to

1. Retrieve details of all books in the library – id, title, name of publisher, authors, number of copies in each branch, etc.
2. Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2017 to Jun 2017.
3. Delete a book in BOOK table. Update the contents of other tables to reflect this data manipulation operation.
4. Partition the BOOK table based on year of publication. Demonstrate its working with a simple query.
5. Create a view of all books and its number of copies that are currently available in the Library.

2 Consider the following schema for Order Database:
SALESMAN(Salesman_id, Name, City, Commission)
CUSTOMER(Customer_id, Cust_Name, City, Grade, Salesman_id)
ORDERS(Ord_No, Purchase_Amt, Ord_Date, Customer_id, Salesman_id)
Write SQL queries to

1. Count the customers with grades above Bangalore's average.

	2. Find the name and numbers of all salesman who had more than one customer. 3. List all the salesman and indicate those who have and don't have customers in their cities (Use UNION operation.) 4. Create a view that finds the salesman who has the customer with the highest order of a day. 5. Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted.
3	Consider the schema for Movie Database: ACTOR(Act_id, Act_Name, Act_Gender) DIRECTOR(Dir_id, Dir_Name, Dir_Phone) MOVIES(Mov_id, Mov_Title, Mov_Year, Mov_Lang, Dir_id) MOVIE_CAST(Act_id, Mov_id, Role) RATING(Mov_id, Rev_Stars) Write SQL queries to 1. List the titles of all movies directed by 'Hitchcock'. 2. Find the movie names where one or more actors acted in two or more movies. 3. List all actors who acted in a movie before 2000 and also in a movie after 2015 (use JOIN operation). 4. Find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title. 5. Update rating of all movies directed by 'Steven Spielberg' to 5.
4	Consider the schema for College Database: STUDENT(USN, SName, Address, Phone, Gender) SEMSEC(SSID, Sem, Sec) CLASS(USN, SSID) SUBJECT(Subcode, Title, Sem, Credits) IAMARKS(USN, Subcode, SSID, Test1, Test2, Test3, FinalIA) Write SQL queries to 1. List all the student details studying in fourth semester 'C' section. 2. Compute the total number of male and female students in each semester and in each section. 3. Create a view of Test1 marks of student USN '1BI15CS101' in all subjects. 4. Calculate the FinalIA (average of best two test marks) and update the corresponding table for all students. 5. Categorize students based on the following criterion: If FinalIA = 17 to 20 then CAT = 'Outstanding' If FinalIA = 12 to 16 then CAT = 'Average' If FinalIA< 12 then CAT = 'Weak' Give these details only for 8th semester A, B, and C section students.
5	Consider the schema for Company Database: EMPLOYEE(SSN, Name, Address, Sex, Salary, SuperSSN, DNo) DEPARTMENT(DNo, DName, MgrSSN, MgrStartDate) DLOCATION(DNo,DLoc) PROJECT(PNo, PName, PLocation, DNo) WORKS_ON(SSN, PNo, Hours) Write SQL queries to 1. Make a list of all project numbers for projects that involve an employee whose last name is 'Scott', either as a worker or as a manager of the department that controls the project.

	2. Show the resulting salaries if every employee working on the 'IoT' project is given a 10 percent raise. 3. Find the sum of the salaries of all employees of the 'Accounts' department, as well as the maximum salary, the minimum salary, and the average salary in this department 4. Retrieve the name of each employee who works on all the projects controlledby department number 5 (use NOT EXISTS operator). 5. For each department that has more than five employees, retrieve the department number and the number of its employees who are making more than Rs. 6,00,000.
Part B: Mini project	
• For any problem selected, write the ER Diagram, apply ER-mapping rules, normalize the relations, and follow the application development process. • Make sure that the application should have five or more tables, at least one trigger and one stored procedure, using suitable frontend tool. • Indicative areas include; health care, education, industry, transport, supply chain, etc.	
Course outcomes: The students should be able to:	
• Create, Update and query on the database. • Demonstrate the working of different concepts of DBMS • Implement, analyze and evaluate the project developed for an application.	
Conduction of Practical Examination: 1. All laboratory experiments from part A are to be included for practical examination. 2. Mini project has to be evaluated for 30 Marks. 3. Report should be prepared in a standard format prescribed for project work. 4. Students are allowed to pick one experiment from the lot. 5. Strictly follow the instructions as printed on the cover page of answer script. 6. Marks distribution: a) Part A: Procedure + Conduction + Viva:10 + 35 +5 =50 Marks b) Part B: Demonstration + Report + Viva voce = 15+10+05 = 30 Marks 7. Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.	

CRYPTOGRAPHY, NETWORK SECURITY AND CYBER LAW
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VI

Subject Code	15CS61	IA Marks	20
Number of Lecture Hours/Week	4	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03

CREDITS – 04

Course objectives: This course will enable students to

- Explain the concepts of Cyber security
- Illustrate key management issues and solutions.
- Familiarize with Cryptography and very essential algorithms
- Introduce cyber Law and ethics to be followed.

Module – 1	**Teaching Hours**
Introduction - Cyber Attacks, Defence Strategies and Techniques, Guiding Principles, Mathematical Background for Cryptography - Modulo Arithmetic's, The Greatest Comma Divisor, Useful Algebraic Structures, Chinese Remainder Theorem, Basics of Cryptography - Preliminaries, Elementary Substitution Ciphers, Elementary Transport Ciphers, Other Cipher Properties, Secret Key Cryptography – Product Ciphers, DES Construction.	**10 Hours**
Module – 2	
Public Key Cryptography and RSA – RSA Operations, Why Does RSA Work?, Performance, Applications, Practical Issues, Public Key Cryptography Standard (PKCS), Cryptographic Hash - Introduction, Properties, Construction, Applications and Performance, The Birthday Attack, Discrete Logarithm and its Applications - Introduction, Diffie-Hellman Key Exchange, Other Applications.	**10 Hours**
Module – 3	
Key Management - Introduction, Digital Certificates, Public Key Infrastructure, Identity–based Encryption, Authentication–I - One way Authentication, Mutual Authentication, Dictionary Attacks, Authentication – II – Centalised Authentication, The Needham-Schroeder Protocol, Kerberos, Biometrics, IPSec-Security at the Network Layer – Security at Different layers: Pros and Cons, IPSec in Action, Internet Key Exchange (IKE) Protocol, Security Policy and IPSEC, Virtual Private Networks, Security at the Transport Layer - Introduction, SSL Handshake Protocol, SSL Record Layer Protocol, OpenSSL.	**10 Hours**
Module – 4	
IEEE 802.11 Wireless LAN Security - Background, Authentication, Confidentiality and Integrity, Viruses, Worms, and Other Malware, Firewalls – Basics, Practical Issues, Intrusion Prevention and Detection - Introduction, Prevention Versus Detection, Types of Instruction Detection Systems, DDoS Attacks Prevention/Detection, Web Service Security – Motivation, Technologies for Web Services, WS- Security, SAML, Other Standards.	**10 Hours**
Module – 5	
IT act aim and objectives, Scope of the act, Major Concepts, Important provisions, Attribution, acknowledgement, and dispatch of electronic records, Secure electronic records and secure digital signatures, Regulation of certifying authorities: Appointment of Controller and Other officers, Digital Signature certificates, Duties of Subscribers, Penalties and adjudication, The cyber	**10 Hours**

regulations appellate tribunal, Offences, Network service providers not to be liable in certain cases, Miscellaneous Provisions.	

Course outcomes: The students should be able to:

- Discuss cryptography and its need to various applications
- Design and develop simple cryptography algorithms
- Understand cyber security and need cyber Law

Question paper pattern:

The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

Text Books:

1. Cryptography, Network Security and Cyber Laws – Bernard Menezes, Cengage Learning, 2010 edition (Chapters-1,3,4,5,6,7,8,9,10,11,12,13,14,15,19(19.1-19.5),21(21.1-21.2),22(22.1-22.4),25

Reference Books:

1. Cryptography and Network Security- Behrouz A Forouzan, Debdeep Mukhopadhyay, Mc-GrawHill, 3rd Edition, 2015
2. Cryptography and Network Security- William Stallings, Pearson Education, 7th Edition
3. Cyber Law simplified- Vivek Sood, Mc-GrawHill, 11th reprint , 2013
4. Cyber security and Cyber Laws, Alfred Basta, Nadine Basta, Mary brown, ravindra kumar, Cengage learning

FILE STRUCTURES
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VI

Subject Code	15IS62	IA Marks	20
Number of Lecture Hours/Week	4	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03

CREDITS – 04

Course objectives: This course will enable students to

- Explain the fundamentals of file structures and their management.
- Measure the performance of different file structures
- Organize different file structures in the memory.
- Demonstrate hashing and indexing techniques.

Module – 1	**Teaching Hours**
Introduction: File Structures: The Heart of the file structure Design, A Short History of File Structure Design, A Conceptual Toolkit; Fundamental File Operations: Physical Files and Logical Files, Opening Files, Closing Files, Reading and Writing, Seeking, Special Characters, The Unix Directory Structure, Physical devices and Logical Files, File-related Header Files, UNIX file System Commands; Secondary Storage and System Software: Disks, Magnetic Tape, Disk versus Tape; CD-ROM: Introduction, Physical Organization, Strengths and Weaknesses; Storage as Hierarchy, A journey of a Byte, Buffer Management, Input /Output in UNIX. **Fundamental File Structure Concepts, Managing Files of Records :** Field and Record Organization, Using Classes to Manipulate Buffers, Using Inheritance for Record Buffer Classes, Managing Fixed Length, Fixed Field Buffers, An Object-Oriented Class for Record Files, Record Access, More about Record Structures, Encapsulating Record Operations in a Single Class, File Access and File Organization.	**10 Hours**
Module – 2	
Organization of Files for Performance, Indexing: Data Compression, Reclaiming Space in files, Internal Sorting and Binary Searching, Keysorting; What is an Index? A Simple Index for Entry-Sequenced File, Using Template Classes in C++ for Object I/O, Object-Oriented support for Indexed, Entry-Sequenced Files of Data Objects, Indexes that are too large to hold in Memory, Indexing to provide access by Multiple keys, Retrieval Using Combinations of Secondary Keys, Improving the Secondary Index structure: Inverted Lists, Selective indexes, Binding.	**10 Hours**
Module – 3	
Consequential Processing and the Sorting of Large Files: A Model for Implementing Cosequential Processes, Application of the Model to a General Ledger Program, Extension of the Model to include Mutiway Merging, A Second Look at Sorting in Memory, Merging as a Way of Sorting Large Files on Disk. **Multi-Level Indexing and B-Trees:** The invention of B-Tree, Statement of the problem, Indexing with Binary Search Trees; Multi-Level Indexing, B-Trees, Example of Creating a B-Tree, An Object-Oriented Representation of B-Trees, B-Tree Methods; Nomenclature, Formal Definition of B-Tree Properties, Worst-case Search Depth, Deletion, Merging and Redistribution, Redistribution during	**10 Hours**

insertion; B* Trees, Buffering of pages; Virtual B-Trees; Variable-length Records and keys.	
Module – 4	
Indexed Sequential File Access and Prefix B + Trees: Indexed Sequential Access, Maintaining a Sequence Set, Adding a Simple Index to the Sequence Set, The Content of the Index: Separators Instead of Keys, The Simple Prefix B+ Tree and its maintenance, Index Set Block Size, Internal Structure of Index Set Blocks: A Variable-order B- Tree, Loading a Simple Prefix B+ Trees, B-Trees, B+ Trees and Simple Prefix B+ Trees in Perspective.	**10 Hours**
Module – 5	
Hashing: Introduction, A Simple Hashing Algorithm, Hashing Functions and Record Distribution, How much Extra Memory should be used?, Collision resolution by progressive overflow, Buckets, Making deletions, Other collision resolution techniques, Patterns of record access. **Extendible Hashing:** How Extendible Hashing Works, Implementation, Deletion, Extendible Hashing Performance, Alternative Approaches.	**10 Hours**
Course outcomes: The students should be able to:	
• Choose appropriate file structure for storage representation. • Identify a suitable sorting technique to arrange the data. • Select suitable indexing and hashing techniques for better performance to a given problem.	
Question paper pattern: The question paper will have TEN questions. There will be TWO questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer FIVE full questions, selecting ONE full question from each module.	
Text Books:	
1. Michael J. Folk, Bill Zoellick, Greg Riccardi: File Structures-An Object Oriented Approach with C++, 3rd Edition, Pearson Education, 1998. (Chapters 1 to 12 excluding 1.4, 1.5, 5.5, 5.6, 8.6, 8.7, 8.8)	
Reference Books:	
1. K.R. Venugopal, K.G. Srinivas, P.M. Krishnaraj: File Structures Using C++, Tata McGraw-Hill, 2008. 2. Scot Robert Ladd: C++ Components and Algorithms, BPB Publications, 1993. 3. Raghu Ramakrishan and Johannes Gehrke: Database Management Systems, 3rd Edition, McGraw Hill, 2003.	

SOFTWARE TESTING
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – V

Subject Code	15IS63	IA Marks	20
Number of Lecture Hours/Week	4	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03

CREDITS – 04

Course objectives: This course will enable students to

- Differentiate the various testing techniques
- Analyze the problem and derive suitable test cases.
- Apply suitable technique for designing of flow graph
- Explain the need for planning and monitoring a process

Module – 1	**Teaching Hours**
Basics of Software Testing: Basic definitions, Software Quality , Requirements, Behaviour and Correctness, Correctness versus Reliability, Testing and Debugging, Test cases, Insights from a Venn diagram, Identifying test cases, Test-generation Strategies, Test Metrics, Error and fault taxonomies , Levels of testing, Testing and Verification, Static Testing. **Problem Statements:** Generalized pseudocode, the triangle problem, the NextDate function, the commission problem, the SATM (Simple Automatic Teller Machine) problem, the currency converter, Saturn windshield wiper **T1:Chapter1, T3:Chapter1, T1:Chapter2.**	**10 Hours**
Module – 2	
Functional Testing: Boundary value analysis, Robustness testing, Worst-case testing, Robust Worst testing for triangle problem, Nextdate problem and commission problem, Equivalence classes, Equivalence test cases for the triangle problem, NextDate function, and the commission problem, Guidelines and observations, Decision tables, Test cases for the triangle problem, NextDate function, and the commission problem, Guidelines and observations. **Fault Based Testing:** Overview, Assumptions in fault based testing, Mutation analysis, Fault-based adequacy criteria, Variations on mutation analysis. **T1: Chapter 5, 6 & 7, T2: Chapter 16**	**10 Hours**
Module – 3	
Structural Testing: Overview, Statement testing, Branch testing, Condition testing , Path testing: DD paths, Test coverage metrics, Basis path testing, guidelines and observations, Data –Flow testing: Definition-Use testing, Slice-based testing, Guidelines and observations**. Test Execution:** Overview of test execution, from test case specification to test cases, Scaffolding, Generic versus specific scaffolding, Test oracles, Self-checks as oracles, Capture and replay **T3:Section 6.2.1, T3:Section 6.2.4, T1:Chapter 9 & 10, T2:Chapter 17**	**10 Hours**
Module – 4	
Process Framework : Basic principles: Sensitivity, redundancy, restriction, partition, visibility, Feedback, the quality process, Planning and monitoring, Quality goals, Dependability properties ,Analysis Testing, Improving the process, Organizational factors. **Planning and Monitoring the Process:** Quality and process, Test and analysis strategies and plans, Risk planning, monitoring the process, Improving the	**10 Hours**

process, the quality team **Documenting Analysis and Test:** Organizing documents, Test strategy document, Analysis and test plan, Test design specifications documents, Test and analysis reports. **T2: Chapter 3 & 4, T2: Chapter 20, T2: Chapter 24.**	
Module – 5	
Integration and Component-Based Software Testing: Overview, Integration testing strategies, Testing components and assemblies. System, Acceptance and Regression Testing: Overview, System testing, Acceptance testing, Usability, Regression testing, Regression test selection techniques, Test case prioritization and selective execution. **Levels of Testing, Integration Testing:** Traditional view of testing levels, Alternative life-cycle models, The SATM system, Separating integration and system testing, A closer look at the SATM system, Decomposition-based, call graph-based, Path-based integrations**.** **T2: Chapter 21 & 22, T1 : Chapter 12 & 13**	**10 Hours**
Course outcomes: The students should be able to:	
• Derive test cases for any given problem • Compare the different testing techniques • Classify the problem into suitable testing model • Apply the appropriate technique for the design of flow graph. • Create appropriate document for the software artefact.	
Question paper pattern: The question paper will have TEN questions. There will be TWO questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer FIVE full questions, selecting ONE full question from each module.	
Text Books:	
1. Paul C. Jorgensen: Software Testing, A Craftsman's Approach, 3rd Edition, Auerbach Publications, 2008. (Listed topics only from Chapters 1, 2, 5, 6, 7, 9, 10, 12, 13) 2. Mauro Pezze, Michal Young: Software Testing and Analysis – Process, Principles and Techniques, Wiley India, 2009. (Listed topics only from Chapters 3, 4, 16, 17, 20,21, 22,24) 3. Aditya P Mathur: Foundations of Software Testing, Pearson Education, 2008.(Listed topics only from Section 1.2 , 1.3, 1.4 ,1.5, 1.8,1.12,6. 2.1,6. 2.4)	
Reference Books:	
1. Software testing Principles and Practices – Gopalaswamy Ramesh, Srinivasan Desikan, 2 nd Edition, Pearson, 2007. 2. Software Testing – Ron Patton, 2nd edition, Pearson Education, 2004. 3. The Craft of Software Testing – Brian Marrick, Pearson Education, 1995. 4. Anirban Basu, Software Quality Assurance, Testing and Metrics, PHI, 2015. 5. Naresh Chauhan, Software Testing, Oxford University press.	

OPERATING SYSTEMS
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VI

Subject Code	15CS64	IA Marks	20
Number of Lecture Hours/Week	4	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03

CREDITS – 04

Course objectives: This course will enable students to

- Introduce concepts and terminology used in OS
- Explain threading and multithreaded systems
- Illustrate process synchronization and concept of Deadlock
- Introduce Memory and Virtual memory management, File system and storage techniques

Module – 1	**Teaching Hours**
Introduction to operating systems, System structures: What operating systems do; Computer System organization; Computer System architecture; Operating System structure; Operating System operations; Process management; Memory management; Storage management; Protection and Security; Distributed system; Special-purpose systems; Computing environments. Operating System Services; User - Operating System interface; System calls; Types of system calls; System programs; Operating system design and implementation; Operating System structure; Virtual machines; Operating System generation; System boot. **Process Management** Process concept; Process scheduling; Operations on processes; Inter process communication	**10 Hours**
Module – 2	
Multi-threaded Programming: Overview; Multithreading models; Thread Libraries; Threading issues. Process Scheduling: Basic concepts; Scheduling Criteria; Scheduling Algorithms; Multiple-processor scheduling; Thread scheduling. **Process Synchronization:** Synchronization: The critical section problem; Peterson's solution; Synchronization hardware; Semaphores; Classical problems of synchronization; Monitors.	**10 Hours**
Module – 3	
Deadlocks : Deadlocks; System model; Deadlock characterization; Methods for handling deadlocks; Deadlock prevention; Deadlock avoidance; Deadlock detection and recovery from deadlock. **Memory Management:** Memory management strategies: Background; Swapping; Contiguous memory allocation; Paging; Structure of page table; Segmentation.	**10 Hours**
Module – 4	
Virtual Memory Management: Background; Demand paging; Copy-on-write; Page replacement; Allocation of frames; Thrashing. **File System, Implementation of File System:** File system: File concept; Access methods; Directory structure; File system mounting; File sharing; Protection: Implementing File system: File system structure; File system implementation; Directory implementation; Allocation methods; Free space management.	**10 Hours**
Module – 5	
Secondary Storage Structures, Protection: Mass storage structures; Disk	**10 Hours**

structure; Disk attachment; Disk scheduling; Disk management; Swap space management. Protection: Goals of protection, Principles of protection, Domain of protection, Access matrix, Implementation of access matrix, Access control, Revocation of access rights, Capability- Based systems. **Case Study: The Linux Operating System:** Linux history; Design principles; Kernel modules; Process management; Scheduling; Memory Management; File systems, Input and output; Inter-process communication.	

Course outcomes: The students should be able to:

- Demonstrate need for OS and different types of OS
- Apply suitable techniques for management of different resources
- Use processor, memory, storage and file system commands
- Realize the different concepts of OS in platform of usage through case studies

Question paper pattern:

The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

Text Books:

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Principles 7th edition, Wiley-India, 2006.

Reference Books

1. Ann McHoes Ida M Fylnn, Understanding Operating System, Cengage Learning, 6th Edition
2. D.M Dhamdhere, Operating Systems: A Concept Based Approach 3rd Ed, McGraw-Hill, 2013.
3. P.C.P. Bhatt, An Introduction to Operating Systems: Concepts and Practice 4th Edition, PHI(EEE), 2014.
4. William Stallings Operating Systems: Internals and Design Principles, 6th Edition, Pearson.

DATA MINING AND DATA WAREHOUSING
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VI

Subject Code	15CS651	IA Marks	20
Number of Lecture Hours/Week	3	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 03

Course objectives: This course will enable students to

- Define multi-dimensional data models.
- Explain rules related to association, classification and clustering analysis.
- Compare and contrast between different classification and clustering algorithms

Module – 1	**Teaching Hours**
Data Warehousing & modeling: Basic Concepts: Data Warehousing: A multitier Architecture, Data warehouse models: Enterprise warehouse, Data mart and virtual warehouse, Extraction, Transformation and loading, Data Cube: A multidimensional data model, Stars, Snowflakes and Fact constellations: Schemas for multidimensional Data models, Dimensions: The role of concept Hierarchies, Measures: Their Categorization and computation, Typical OLAP Operations.	**8 Hours**
Module – 2	
Data warehouse implementation& Data mining: Efficient Data Cube computation: An overview, Indexing OLAP Data: Bitmap index and join index, Efficient processing of OLAP Queries, OLAP server Architecture ROLAP versus MOLAP Versus HOLAP. **:** Introduction: What is data mining, Challenges, Data Mining Tasks, Data: Types of Data, Data Quality, Data Preprocessing, Measures of Similarity and Dissimilarity,	**8 Hours**
Module – 3	
Association Analysis: Association Analysis: Problem Definition, Frequent Item set Generation, Rule generation. Alternative Methods for Generating Frequent Item sets, FP-Growth Algorithm, Evaluation of Association Patterns.	**8 Hours**
Module – 4	
Classification : Decision Trees Induction, Method for Comparing Classifiers, Rule Based Classifiers, Nearest Neighbor Classifiers, Bayesian Classifiers.	**8 Hours**
Module – 5	
Clustering Analysis: Overview, K-Means, Agglomerative Hierarchical Clustering, DBSCAN, Cluster Evaluation, Density-Based Clustering, Graph-Based Clustering, Scalable Clustering Algorithms.	**8 Hours**

Course outcomes: The students should be able to:

- Identify data mining problems and implement the data warehouse
- Write association rules for a given data pattern.
- Choose between classification and clustering solution.

Question paper pattern:
The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.

The students will have to answer FIVE full questions, selecting ONE full question from each module.
Text Books:
1. Pang-Ning Tan, Michael Steinbach, Vipin Kumar: Introduction to Data Mining, Pearson, First impression,2014. 2. Jiawei Han, Micheline Kamber, Jian Pei: Data Mining -Concepts and Techniques, 3rd Edition, Morgan Kaufmann Publisher, 2012.
Reference Books:
1. Sam Anahory, Dennis Murray: Data Warehousing in the Real World, Pearson,Tenth Impression,2012. 2. Michael.J.Berry,Gordon.S.Linoff: Mastering Data Mining , Wiley Edition, second edtion,2012.

SYSTEM SOFTWARE
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VI

Subject Code	15IS652	IA Marks	20
Number of Lecture Hours/Week	3	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 03

Course objectives: This course will enable students to

- Define System Software such as Assemblers, Loaders, Linkers and Macroprocessors
- Familiarize with source file, object file and executable file structures and libraries
- Describe the front-end and back-end phases of compiler and their importance to students

Module – 1	**Teaching Hours**
Introduction to System Software, Machine Architecture of SIC and SIC/XE. **Assemblers:** Basic assembler functions, machine dependent assembler features, machine independent assembler features, assembler design options. **Macroprocessors:** Basic macro processor functions, machine independent macro processor features, Macro processor design options, implementation examples **Text book 1: Chapter 1: (1.1-1.3.2), Chapter2: 2.1- 2.4 ,Chapter4**	**08 Hours**
Module – 2	
Loaders and Linkers: Basic Loader Functions**,** Design of an absolute loader, a simple Bootstrap loader, Machine-dependent loader features-relocation, program linking, algorithm and data structures for a linking loader, Machine –independent loader features-automatic library search, Loader options, loader design options-linkage editor, dynamic linkage, bootstrap loaders, implementation examples-MS DOS linker. **Text book 1 : Chapter 3**	**08 Hours**
Module – 3	
System File and Library Structure: Introduction, Library And File Organization, Design Of A Record Source Program File Structure, Object Code, Object File, Object File Structure, Executable File, Executable File Structure, Libraries, Image File Structure. **Object Code translators:** introduction, binary code translators, object code translators, translation process, hybrid method, applications **Reference 1: chapter 5 and chapter 15**	**08 Hours**
Module – 4	
Lexical Analysis: Introduction, Alphabets And Tokens In Computer Languages, Representation, Token Recognition And Finite Automata, Implementation, Error Recovery. **Text book 2: Chapter 1(1.1-1.5), Chapter 3(3.1-3.5)**	**08 Hours**
Module – 5	
Syntax Analysis: Introduction, Role Of Parsers, Context Free Grammars, Top Down Parsers, Bottom-Up Parsers, Operator-Precedence Parsing **Text book 2: Chapter 4 (4.1 – 4.6)**	**08 Hours**

Course outcomes: The students should be able to:

- Explain system software such as assemblers, loaders, linkers and macroprocessors
- Design and develop lexical analyzers, parsers and code generators
- Utilize lex and yacc tools for implementing different concepts of system software

Question paper pattern:

The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

Text Books:

1. System Software by Leland. L. Beck, D Manjula, 3rd edition, 2012
2. Compilers-Principles, Techniques and Tools by Alfred V Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman. Pearson, 2nd edition, 2007

Reference Books:

1. Systems programming – Srimanta Pal , Oxford university press, 2016
2. System software and operating system by D. M. Dhamdhere TMG
3. Compiler Design, K Muneeswaran, Oxford University Press 2013.
4. System programming and Compiler Design, K C Louden, Cengage Learning

OPERATIONS RESEARCH
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VI

Subject Code	15CS653	IA Marks	20
Number of Lecture Hours/Week	3	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 03

Course objectives: This course will enable students to

- Formulate optimization problem as a linear programming problem.
- Solve optimization problems using simplex method.
- Formulate and solve transportation and assignment problems.
- Apply game theory for decision making problems.

Module – 1	**Teaching Hours**
Introduction, Linear Programming: Introduction: The origin, nature and impact of OR; Defining the problem and gathering data; Formulating a mathematical model; Deriving solutions from the model; Testing the model; Preparing to apply the model; Implementation . **Introduction to Linear Programming Problem (LPP):** Prototype example, Assumptions of LPP, Formulation of LPP and Graphical method various examples.	**8 Hours**
Module – 2	
Simplex Method – 1: The essence of the simplex method; Setting up the simplex method; Types of variables, Algebra of the simplex method; the simplex method in tabular form; Tie breaking in the simplex method, Big M method, Two phase method.	**8 Hours**
Module – 3	
Simplex Method – 2: Duality Theory - The essence of duality theory, Primal dual relationship, conversion of primal to dual problem and vice versa. The dual simplex method.	**8 Hours**
Module – 4	
Transportation and Assignment Problems: The transportation problem, Initial Basic Feasible Solution (IBFS) by North West Corner Rule method, Matrix Minima Method, Vogel's Approximation Method. Optimal solution by Modified Distribution Method (MODI). The Assignment problem; A Hungarian algorithm for the assignment problem. Minimization and Maximization varieties in transportation and assignment problems.	**8 Hours**
Module – 5	
Game Theory: Game Theory: The formulation of two persons, zero sum games; saddle point, maximin and minimax principle, Solving simple games- a prototype example; Games with mixed strategies; Graphical solution procedure. **Metaheuristics:** The nature of Metaheuristics, Tabu Search, Simulated Annealing, Genetic Algorithms.	**8 Hours**

Course outcomes: The students should be able to:

- Select and apply optimization techniques for various problems.
- Model the given problem as transportation and assignment problem and solve.
- Apply game theory for decision support system.

Question paper pattern:
The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

Text Books:

1. D.S. Hira and P.K. Gupta, Operations Research, (Revised Edition), Published by S. Chand & Company Ltd, 2014

Reference Books:

1. S Kalavathy, Operation Research, Vikas Publishing House Pvt Limited, 01-Aug-2002
2. S D Sharma, Operation Research, Kedar Nath Ram Nath Publishers.

DISTRIBUTED COMPUTING SYSTEM
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VI

Subject Code	15CS654	IA Marks	20
Number of Lecture Hours/Week	3	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 03

Course objectives: This course will enable students to

- Explain distributed system, their characteristics, challenges and system models.
- Describe IPC mechanisms to communicate between distributed objects
- Illustrate the operating system support and File Service architecture in a distributed system
- Analyze the fundamental concepts, algorithms related to synchronization.

Module – 1	**Teaching Hours**
Characterization of Distributed Systems: Introduction, Examples of DS, Resource sharing and the Web, Challenges **System Models:** Architectural Models, Fundamental Models	**8 Hours**
Module – 2	
Inter Process Communication: Introduction, API for Internet Protocols, External Data Representation and Marshalling, Client – Server Communication, Group Communication **Distributed Objects and RMI:** Introduction, Communication between Distributed Objects, RPC, Events and Notifications	**8 Hours**
Module – 3	
Operating System Support: Introduction, The OS layer, Protection, Processes and Threads, Communication and Invocation , Operating system architecture **Distributed File Systems:** Introduction, File Service architecture, Sun Network File System	**8 Hours**
Module – 4	
Time and Global States: Introduction, Clocks, events and process status, Synchronizing physical clocks, Logical time and logical clocks, Global states **Coordination and Agreement:** Introduction, Distributed mutual exclusion, Elections	**8 Hours**
Module – 5	
Distributed Transactions: Introduction, Flat and nested distributed transactions, Atomic commit protocols, Concurrency control in distributed transactions, distributed deadlocks	**8 Hours**

Course outcomes: The students should be able to:

- Explain the characteristics of a distributed system along with its and design challenges
- Illustrate the mechanism of IPC between distributed objects
- Describe the distributed file service architecture and the important characteristics of SUN NFS.
- Discuss concurrency control algorithms applied in distributed transactions

Question paper pattern:
The question paper will have TEN questions.

There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

Text Books:

1. George Coulouris, Jean Dollimore and Tim Kindberg: Distributed Systems – Concepts and Design, 5th Edition, Pearson Publications, 2009

Reference Books:

1. Andrew S Tanenbaum: Distributed Operating Systems, 3rd edition, Pearson publication, 2007
2. Ajay D. Kshemkalyani and Mukesh Singhal, Distributed Computing: Principles, Algorithms and Systems, Cambridge University Press, 2008
3. Sunita Mahajan, Seema Shan, " Distributed Computing", Oxford University Press,2015

SOFTWARE TESTING LABORATORY
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VI

Subject Code	15ISL67	IA Marks	20
Number of Lecture Hours/Week	01I + 02P	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 02

Course objectives: This course will enable students to

- Analyse the requirements for the given problem statement
- Design and implement various solutions for the given problem
- Employ various design strategies for problem solving.
- Construct control flow graphs for the solution that is implemented
- Create appropriate document for the software artefact

Description (If any):

Design, develop, and implement the specified algorithms for the following problems using any language of your choice under LINUX /Windows environment.

Lab Experiments:

1. Design and develop a program in a language of your choice to solve the triangle problem defined as follows: Accept three integers which are supposed to be the three sides of a triangle and determine if the three values represent an equilateral triangle, isosceles triangle, scalene triangle, or they do not form a triangle at all. Assume that the upper limit for the size of any side is 10. Derive test cases for your program based on boundary-value analysis, execute the test cases and discuss the results.

2. Design, develop, code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of boundary value testing, derive different test cases, execute these test cases and discuss the test results.

3. Design, develop, code and run the program in any suitable language to implement the NextDate function. Analyze it from the perspective of boundary value testing, derive different test cases, execute these test cases and discuss the test results.

4. Design and develop a program in a language of your choice to solve the triangle problem defined as follows: Accept three integers which are supposed to be the three sides of a triangle and determine if the three values represent an equilateral triangle, isosceles triangle, scalene triangle, or they do not form a triangle at all. Assume that the upper limit for the size of any side is 10. Derive test cases for your program based on equivalence class partitioning, execute the test cases and discuss the results.

5. Design, develop, code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of equivalence class testing, derive different test cases, execute these test cases and discuss the test results.

6. Design, develop, code and run the program in any suitable language to implement the NextDate function. Analyze it from the perspective of equivalence class value testing, derive different test cases, execute these test cases and discuss the test results.

7. Design and develop a program in a language of your choice to solve the triangle problem defined as follows: Accept three integers which are supposed to be the three sides of a triangle and determine if the three values represent an equilateral triangle,

isosceles triangle, scalene triangle, or they do not form a triangle at all. Derive test cases for your program based on decision-table approach, execute the test cases and discuss the results.

8. Design, develop, code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of decision table-based testing, derive different test cases, execute these test cases and discuss the test results.
9. Design, develop, code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of dataflow testing, derive different test cases, execute these test cases and discuss the test results.
10. Design, develop, code and run the program in any suitable language to implement the binary search algorithm. Determine the basis paths and using them derive different test cases, execute these test cases and discuss the test results.
11. Design, develop, code and run the program in any suitable language to implement the quicksort algorithm. Determine the basis paths and using them derive different test cases, execute these test cases and discuss the test results.
12. Design, develop, code and run the program in any suitable language to implement an absolute letter grading procedure, making suitable assumptions. Determine the basis paths and using them derive different test cases, execute these test cases and discuss the test results

Study Experiment / Project:

1. Design, develop, code and run the program in any suitable language to solve the triangle problem. Analyze it from the perspective of dataflow testing, derive different test cases, execute these test cases and discuss the test results.

2. Design, develop, code and run the program in any suitable language to solve the Nextdate problem. Analyze it from the perspective of decision table-based testing, derive different test cases, execute these test cases and discuss the test results.

Course outcomes: The students should be able to:

- List out the requirements for the given problem
- Design and implement the solution for given problem in any programming language(C,C++,JAVA)
- Derive test cases for any given problem
- Apply the appropriate technique for the design of flow graph.
- Create appropriate document for the software artefact.

Conduction of Practical Examination:

1. All laboratory experiments are to be included for practical examination.
2. Students are allowed to pick one experiment from the lot.
3. Strictly follow the instructions as printed on the cover page of answer script for breakup of marks
4. Procedure + Conduction + Viva: 35 + 35 + 10 (80)
5. Change of experiment is allowed only once and marks allotted to the procedure part to be made zero

FILE STRUCTURES LABORATORY WITH MINI PROJECT
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VI

Subject Code	15ISL68	IA Marks	20
Number of Lecture Hours/Week	01I + 02P	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 02

Course objectives: This course will enable students to

- Apply the concepts of Unix IPC to implement a given function.
- Measure the performance of different file structures
- Write a program to manage operations on given file system.
- Demonstrate hashing and indexing techniques

Description (If any):

Design, develop, and implement the following programs

Lab Experiments:

PART A

1. Write a program to read series of names, one per line, from standard input and write these names spelled in reverse order to the standard output using I/O redirection and pipes. Repeat the exercise using an input file specified by the user instead of the standard input and using an output file specified by the user instead of the standard output.
2. Write a program to read and write student objects with fixed-length records and the fields delimited by "|". Implement pack ( ), unpack ( ), modify ( ) and search ( ) methods.
3. Write a program to read and write student objects with Variable - Length records using any suitable record structure. Implement pack ( ), unpack ( ), modify ( ) and search ( ) methods.
4. Write a program to write student objects with Variable - Length records using any suitable record structure and to read from this file a student record using RRN.
5. Write a program to implement simple index on primary key for a file of student objects. Implement add ( ), search ( ), delete ( ) using the index.
6. Write a program to implement index on secondary key, the name, for a file of student objects. Implement add ( ), search ( ), delete ( ) using the secondary index.
7. Write a program to read two lists of names and then match the names in the two lists using Consequential Match based on a single loop. Output the names common to both the lists.
8. Write a program to read k Lists of names and merge them using k-way merge algorithm with k = 8.

Part B --- Mini project:

Student should develop mini project on the topics mentioned below or similar applications **Document processing, transaction management, indexing and hashing, buffer management, configuration management. Not limited to these.**

Course outcomes: The students should be able to:

- Implement operations related to files
- Apply the concepts of file system to produce the given application.
- Evaluate performance of various file systems on given parameters.

Conduction of Practical Examination:

1. All laboratory experiments from part A are to be included for practical examination.
2. Mini project has to be evaluated for 30 Marks as per 6(b).
3. Report should be prepared in a standard format prescribed for project work.
4. Students are allowed to pick one experiment from the lot.
5. Strictly follow the instructions as printed on the cover page of answer script.
6. Marks distribution:
   a) Part A: Procedure + Conduction + Viva:10 + 35 +5 =50 Marks
   b) Part B: Demonstration + Report + Viva voce = 15+10+05 = 30 Marks
7. Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.

WEB TECHNOLOGY AND ITS APPLICATIONS
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VII

Subject Code	**15CS71**	IA Marks	20
Number of Lecture Hours/Week	04	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03

CREDITS – 04

Course Objectives: This course will enable students to

- Illustrate the Semantic Structure of HTML and CSS
- Compose forms and tables using HTML and CSS
- Design Client-Side programs using JavaScript and Server-Side programs using PHP
- Infer Object Oriented Programming capabilities of PHP
- Examine JavaScript frameworks such as jQuery and Backbone

Module – 1	**Teaching Hours**
Introduction to HTML, What is HTML and Where did it come from?, HTML Syntax, Semantic Markup, Structure of HTML Documents, Quick Tour of HTML Elements, HTML5 Semantic Structure Elements, Introduction to CSS, What is CSS, CSS Syntax, Location of Styles, Selectors, The Cascade: How Styles Interact, The Box Model, CSS Text Styling.	**10 Hours**
Module – 2	
HTML Tables and Forms, Introducing Tables, Styling Tables, Introducing Forms, Form Control Elements, Table and Form Accessibility, Microformats, Advanced CSS: Layout, Normal Flow, Positioning Elements, Floating Elements, Constructing Multicolumn Layouts, Approaches to CSS Layout, Responsive Design, CSS Frameworks.	**10 Hours**
Module – 3	
JavaScript: Client-Side Scripting, What is JavaScript and What can it do?, JavaScript Design Principles, Where does JavaScript Go?, Syntax, JavaScript Objects, The Document Object Model (DOM), JavaScript Events, Forms, Introduction to Server-Side Development with PHP, What is Server-Side Development, A Web Server's Responsibilities, Quick Tour of PHP, Program Control, Functions	**10 Hours**
Module – 4	
PHP Arrays and Superglobals, Arrays, $_GET and $_POST Superglobal Arrays, $_SERVER Array, $_Files Array, Reading/Writing Files, PHP Classes and Objects, Object-Oriented Overview, Classes and Objects in PHP, Object Oriented Design, Error Handling and Validation, What are Errors and Exceptions?, PHP Error Reporting, PHP Error and Exception Handling	**10 Hours**
Module – 5	
Managing State, The Problem of State in Web Applications, Passing Information via Query Strings, Passing Information via the URL Path, Cookies, Serialization, Session State, HTML5 Web Storage, Caching, Advanced JavaScript and jQuery, JavaScript Pseudo-Classes, jQuery Foundations, AJAX, Asynchronous File Transmission, Animation, Backbone MVC Frameworks, XML Processing and Web Services, XML Processing, JSON, Overview of Web Services.	**10 Hours**

Course Outcomes: After studying this course, students will be able to

- Adapt HTML and CSS syntax and semantics to build web pages.

- Construct and visually format tables and forms using HTML and CSS
- Develop Client-Side Scripts using JavaScript and Server-Side Scripts using PHP to generate and display the contents dynamically.
- Appraise the principles of object oriented development using PHP
- Inspect JavaScript frameworks like jQuery and Backbone which facilitates developer to focus on core features.

Question paper pattern:

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Randy Connolly, Ricardo Hoar, **"Fundamentals of Web Development",** 1st Edition, Pearson Education India. (**ISBN:**978-9332575271)

Reference Books:

1) Robin Nixon, **"Learning PHP, MySQL &JavaScript with jQuery, CSS and HTML5",** 4th Edition, O'Reilly Publications, 2015. (**ISBN:**978-9352130153)
2) Luke Welling, Laura Thomson, **"PHP and MySQL Web Development",** 5th Edition, Pearson Education, 2016. (**ISBN:**978-9332582736)
3) Nicholas C Zakas, **"Professional JavaScript for Web Developers",** 3rd Edition, Wrox/Wiley India, 2012. (**ISBN:**978-8126535088)
4) David Sawyer Mcfarland, **"JavaScript & jQuery: The Missing Manual",** 1st Edition, O'Reilly/Shroff Publishers & Distributors Pvt Ltd, 2014 (**ISBN:**978-9351108078)
5) Zak Ruvalcaba Anne Boehm, **"Murach's HTML5 and CSS3"**, 3rd Edition, Murachs/Shroff Publishers & Distributors Pvt Ltd, 2016. (**ISBN:**978-9352133246)

SOFTWARE ARCHITECTURE AND DESIGN PATTERNS
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VII

Subject Code	15IS72	IA Marks	20
Number of Lecture Hours/Week	4	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03

CREDITS – 04

Course objectives: This course will enable students to

- Learn How to add functionality to designs while minimizing complexity.
- What code qualities are required to maintain to keep code flexible?
- To Understand the common design patterns.
- To explore the appropriate patterns for design problems

Module – 1	**Teaching Hours**
Introduction: what is a design pattern? describing design patterns, the catalog of design pattern, organizing the catalog, how design patterns solve design problems, how to select a design pattern, how to use a design pattern. What is object-oriented development? , key concepts of object oriented design other related concepts, benefits and drawbacks of the paradigm	**10 Hours**
Module – 2	
Analysis a System: overview of the analysis phase, stage 1: gathering the requirements functional requirements specification, defining conceptual classes and relationships, using the knowledge of the domain. Design and Implementation, discussions and further reading.	**10 Hours**
Module – 3	
Design Pattern Catalog: Structural patterns, Adapter, bridge, composite, decorator, facade, flyweight, proxy.	**10 Hours**
Module – 4	
Interactive systems and the MVC architecture: Introduction , The MVC architectural pattern, analyzing a simple drawing program , designing the system, designing of the subsystems, getting into implementation , implementing undo operation , drawing incomplete items, adding a new feature , pattern based solutions.	**10 Hours**
Module – 5	
Designing with Distributed Objects: Client server system, java remote method invocation, implementing an object oriented system on the web (discussions and further reading) a note on input and output, selection statements, loops arrays.	**10 Hours**

Course outcomes: The students should be able to:

- Design and implement codes with higher performance and lower complexity
- Be aware of code qualities needed to keep code flexible
- Experience core design principles and be able to assess the quality of a design with respect to these principles.
- Capable of applying these principles in the design of object oriented systems.
- Demonstrate an understanding of a range of design patterns. Be capable of comprehending a design presented using this vocabulary.
- Be able to select and apply suitable patterns in specific contexts

Question paper pattern:

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Object-oriented analysis, design and implementation, brahma dathan, sarnath rammath, universities press,2013
2. Design patterns, erich gamma, Richard helan, Ralph johman , john vlissides ,PEARSON Publication,2013.

Reference Books:

1. Frank Bachmann, ReginеMeunier, Hans Rohnert “Pattern Oriented Software Architecture” –Volume 1, 1996.
2. William J Brown et al., "Anti-Patterns: Refactoring Software, Architectures and Projects in Crisis", John Wiley, 1998.

MACHINE LEARNING
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VII

Subject Code	15CS73	IA Marks	20
Number of Lecture Hours/Week	03	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03

CREDITS – 04

Course Objectives: This course will enable students to

- Define machine learning and problems relevant to machine learning.
- Differentiate supervised, unsupervised and reinforcement learning
- Apply neural networks, Bayes classifier and k nearest neighbor, for problems appear in machine learning.
- Perform statistical analysis of machine learning techniques.

Module – 1	**Teaching Hours**
Introduction: Well posed learning problems, Designing a Learning system, Perspective and Issues in Machine Learning. **Concept Learning:** Concept learning task, Concept learning as search, Find-S algorithm, Version space, Candidate Elimination algorithm, Inductive Bias. **Text Book1, Sections: 1.1 – 1.3, 2.1-2.5, 2.7**	**10 Hours**
Module – 2	
Decision Tree Learning: Decision tree representation, Appropriate problems for decision tree learning, Basic decision tree learning algorithm, hypothesis space search in decision tree learning, Inductive bias in decision tree learning, Issues in decision tree learning. **Text Book1, Sections: 3.1-3.7**	**10 Hours**
Module – 3	
Artificial Neural Networks: Introduction, Neural Network representation, Appropriate problems, Perceptrons, Backpropagation algorithm. **Text book 1, Sections: 4.1 – 4.6**	**08 Hours**
Module – 4	
Bayesian Learning: Introduction, Bayes theorem, Bayes theorem and concept learning, ML and LS error hypothesis, ML for predicting probabilities, MDL principle, Naive Bayes classifier, Bayesian belief networks, EM algorithm **Text book 1, Sections: 6.1 – 6.6, 6.9, 6.11, 6.12**	**10 Hours**
Module – 5	
Evaluating Hypothesis: Motivation, Estimating hypothesis accuracy, Basics of sampling theorem, General approach for deriving confidence intervals, Difference in error of two hypothesis, Comparing learning algorithms. **Instance Based Learning:** Introduction, k-nearest neighbor learning, locally weighted regression, radial basis function, cased-based reasoning, **Reinforcement Learning:** Introduction, Learning Task, Q Learning **Text book 1, Sections: 5.1-5.6, 8.1-8.5, 13.1-13.3**	**12 Hours**

Course Outcomes: After studying this course, students will be able to

- Identify the problems for machine learning. And select the either supervised,

unsupersvised or reinforcement learning.

- Explain theory of probability and statistics related to machine learning
- Investigate concept learning, ANN, Bayes classifier, k nearest neighbor, Q,

Question paper pattern:

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Tom M. Mitchell, Machine Learning**,** India Edition 2013, McGraw Hill Education.

Reference Books:

1. Trevor Hastie, Robert Tibshirani, Jerome Friedman, h The Elements of Statistical Learning, 2nd edition, springer series in statistics.
2. Ethem Alpaydın, Introduction to machine learning, second edition, MIT press.

NATURAL LANGUAGE PROCESSING
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VII

Subject Code	15CS741	IA Marks	20
Number of Lecture Hours/Week	3	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 03

Course objectives: This course will enable students to

- Learn the techniques in natural language processing.
- Be familiar with the natural language generation.
- Be exposed to Text Mining.
- Understand the information retrieval techniques

Module – 1	**Teaching Hours**
Overview and language modeling: Overview: Origins and challenges of NLP-Language and Grammar-Processing Indian Languages- NLP Applications-Information Retrieval. Language Modeling: Various Grammar- based Language Models-Statistical Language Model.	**8 Hours**
Module – 2	
Word level and syntactic analysis: Word Level Analysis: Regular Expressions-Finite-State Automata-Morphological Parsing-Spelling Error Detection and correction-Words and Word classes-Part-of Speech Tagging. Syntactic Analysis: Context-free Grammar-Constituency- Parsing-Probabilistic Parsing.	**8 Hours**
Module – 3	
Extracting Relations from Text: From Word Sequences to Dependency Paths: Introduction, Subsequence Kernels for Relation Extraction, A Dependency-Path Kernel for Relation Extraction and Experimental Evaluation. **Mining Diagnostic Text Reports by Learning to Annotate Knowledge Roles:** Introduction, Domain Knowledge and Knowledge Roles, Frame Semantics and Semantic Role Labeling, Learning to Annotate Cases with Knowledge Roles and Evaluations. **A Case Study in Natural Language Based Web Search:** InFact System Overview, The GlobalSecurity.org Experience.	**8 Hours**
Module – 4	
Evaluating Self-Explanations in iSTART: Word Matching, Latent Semantic Analysis, and Topic Models: Introduction, iSTART: Feedback Systems, iSTART: Evaluation of Feedback Systems, **Textual Signatures: Identifying Text-Types Using Latent Semantic Analysis to Measure the Cohesion of Text Structures:** Introduction, Cohesion, Coh-Metrix, Approaches to Analyzing Texts, Latent Semantic Analysis, Predictions, Results of Experiments. **Automatic Document Separation: A Combination of Probabilistic Classification and Finite-State Sequence Modeling:** Introduction, Related Work, Data Preparation, Document Separation as a Sequence Mapping Problem, Results. **Evolving Explanatory Novel Patterns for Semantically-Based Text Mining:** Related Work, A Semantically Guided Model for Effective Text Mining.	**8 Hours**

Module – 5

INFORMATION RETRIEVAL AND LEXICAL RESOURCES: Information Retrieval: Design features of Information Retrieval Systems-Classical, Non classical, Alternative Models of Information Retrieval – valuation Lexical Resources: World Net-Frame Net- Stemmers-POS Tagger- Research Corpora.	**8 Hours**

Course outcomes: The students should be able to:

- Analyze the natural language text.
- Generate the natural language.
- Do Text mining.
- Apply information retrieval techniques.

Question paper pattern:

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Tanveer Siddiqui, U.S. Tiwary, "Natural Language Processing and Information Retrieval", Oxford University Press, 2008.
2. Anne Kao and Stephen R. Poteet (Eds), "Natural LanguageProcessing and Text Mining", Springer-Verlag London Limited 2007.

Reference Books:

1. Daniel Jurafsky and James H Martin, "Speech and Language Processing: Anintroduction to Natural Language Processing, Computational Linguistics and SpeechRecognition", 2nd Edition, Prentice Hall, 2008.
2. James Allen, "Natural Language Understanding", 2nd edition, Benjamin/Cummingspublishing company, 1995.
3. Gerald J. Kowalski and Mark.T. Maybury, "Information Storage and Retrieval systems", Kluwer academic Publishers, 2000.

CLOUD COMPUTING AND ITS APPLICATIONS
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VII

Subject Code	15CS742	IA Marks	20
Number of Lecture Hours/Week	3	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 03

Course objectives: This course will enable students to

- Explain the fundamentals of cloud computing
- Illustrate the cloud application programming and aneka platform
- Contrast different cloud platforms used in industry

Module – 1	**Teaching Hours**
Introduction ,Cloud Computing at a Glance, The Vision of Cloud Computing, Defining a Cloud, A Closer Look, Cloud Computing Reference Model, Characteristics and Benefits, Challenges Ahead, Historical Developments, Distributed Systems, Virtualization, Web 2.0, Service-Oriented Computing, Utility-Oriented Computing, Building Cloud Computing Environments, Application Development, Infrastructure and System Development, Computing Platforms and Technologies, Amazon Web Services (AWS), Google AppEngine, Microsoft Azure, Hadoop, Force.com and Salesforce.com, Manjrasoft Aneka Virtualization, Introduction, Characteristics of Virtualized, Environments Taxonomy of Virtualization Techniques, Execution Virtualization, Other Types of Virtualization, Virtualization and Cloud Computing, Pros and Cons of Virtualization, Technology Examples Xen: Paravirtualization, VMware: Full Virtualization, Microsoft Hyper-V	**8 Hours**
Module – 2	
Cloud Computing Architecture, Introduction, Cloud Reference Model, Architecture, Infrastructure / Hardware as a Service, Platform as a Service, Software as a Service, Types of Clouds, Public Clouds, Private Clouds, Hybrid Clouds, Community Clouds, Economics of the Cloud, Open Challenges, Cloud Definition, Cloud Interoperability and Standards Scalability and Fault Tolerance Security, Trust, and Privacy Organizational Aspects Aneka: Cloud Application Platform, Framework Overview, Anatomy of the Aneka Container, From the Ground Up: Platform Abstraction Layer, Fabric Services, foundation Services, Application Services, Building Aneka Clouds, Infrastructure Organization, Logical Organization, Private Cloud Deployment Mode, Public Cloud Deployment Mode, Hybrid Cloud Deployment Mode, Cloud Programming and Management, Aneka SDK, Management Tools	**8 Hours**
Module – 3	
Concurrent Computing: Thread Programming, Introducing Parallelism for Single Machine Computation, Programming Applications with Threads, What is a Thread?, Thread APIs, Techniques for Parallel Computation with Threads, Multithreading with Aneka, Introducing the Thread Programming Model, Aneka Thread vs. Common Threads, Programming Applications with Aneka Threads, Aneka Threads Application Model, Domain Decomposition: Matrix	**8 Hours**

Multiplication, Functional Decomposition: Sine, Cosine, and Tangent. High-Throughput Computing: Task Programming, Task Computing, Characterizing a Task, Computing Categories, Frameworks for Task Computing, Task-based Application Models, Embarrassingly Parallel Applications, Parameter Sweep Applications, MPI Applications, Workflow Applications with Task Dependencies, Aneka Task-Based Programming, Task Programming Model, Developing Applications with the Task Model, Developing Parameter Sweep Application, Managing Workflows.	
Module – 4	
Data Intensive Computing: Map-Reduce Programming, What is Data-Intensive Computing?, Characterizing Data-Intensive Computations, Challenges Ahead, Historical Perspective, Technologies for Data-Intensive Computing, Storage Systems, Programming Platforms, Aneka MapReduce Programming, Introducing the MapReduce Programming Model, Example Application	**8 Hours**
Module – 5	
Cloud Platforms in Industry, Amazon Web Services, Compute Services, Storage Services, Communication Services, Additional Services, Google AppEngine, Architecture and Core Concepts, Application Life-Cycle, Cost Model, Observations, Microsoft Azure, Azure Core Concepts, SQL Azure, Windows Azure Platform Appliance. Cloud Applications Scientific Applications, Healthcare: ECG Analysis in the Cloud, Biology: Protein Structure Prediction, Biology: Gene Expression Data Analysis for Cancer Diagnosis, Geoscience: Satellite Image Processing, Business and Consumer Applications, CRM and ERP, Productivity, Social Networking, Media Applications, Multiplayer Online Gaming.	**8 Hours**
Course outcomes: The students should be able to:	
• Explain cloud computing, virtualization and classify services of cloud computing • Illustrate architecture and programming in cloud • Describe the platforms for development of cloud applications and List the application of cloud.	
Question paper pattern: The question paper will have ten questions. There will be 2 questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer 5 full questions, selecting one full question from each module.	
Text Books: 1. Rajkumar Buyya, Christian Vecchiola, and Thamarai Selvi Mastering Cloud. Computing McGraw Hill Education	
Reference Books:	
1. Dan C. Marinescu, Cloud Computing Theory and Practice, Morgan Kaufmann, Elsevier 2013.	

INFORMATION AND NETWORK SECURITY
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VII

Subject Code	15CS743	IA Marks	20
Number of Lecture Hours/Week	3	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 03

Course objectives: This course will enable students to

- Analyze the cryptographic processes.
- Summarize the digital security process.
- Indicate the location of a security process in the given system

Module – 1	**Teaching Hours**
Introduction. How to Speak Crypto. Classic Crypto. Simple Substitution Cipher. Cryptanalysis of a Simple Substitution. Definition of Secure. Double Transposition Cipher. One-time Pad. Project VENONA. Codebook Cipher. Ciphers of the Election of 1876. Modern Crypto History. Taxonomy of Cryptography. Taxonomy of Cryptanalysis.	**8 Hours**
Module – 2.	
What is a Hash Function? The Birthday Problem.Non-cryptographic Hashes. Tiger Hash. HMAC. Uses of Hash Functions. Online Bids. Spam Reduction. Other Crypto-Related Topics. Secret Sharing. Key Escrow. Random Numbers. Texas Hold 'em Poker. Generating Random Bits. Information Hiding.	**8 Hours**
Module – 3	
Random number generation Providing freshness Fundamentals of entity authentication Passwords Dynamic password schemes Zero-knowledge mechanisms Further reading Cryptographic Protocols Protocol basics From objectives to a protocol Analysing a simple protocol Authentication and key establishment protocols	**8 Hours**
Module – 4	
Key management fundamentals Key lengths and lifetimes Key generation Key establishment Key storage Key usage Governing key management Public-Key Management Certification of public keys The certificate lifecycle Public-key management models Alternative approaches	**8 Hours**
Module – 5	
Cryptographic Applications Cryptography on the Internet Cryptography for wireless local area networks Cryptography for mobile telecommunications Cryptography for secure payment card transactions Cryptography for video broadcasting Cryptography for identity cards Cryptography for home users	**8 Hours**

Course outcomes: The students should be able to:

- Analyze the Digitals security lapses
- Illustrate the need of key management

Question paper pattern:
The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Information Security: Principles and Practice, 2nd Edition by Mark Stamp Wiley
2. Everyday Cryptography: Fundamental Principles and Applications Keith M. Martin Oxford Scholarship Online: December 2013

Reference Books:

1. Applied Cryptography Protocols, Algorithms, and Source Code in C by Bruce Schneier

UNIX SYSTEM PROGRAMMING
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VII

Subject Code	15CS744	IA Marks	20
Number of Lecture Hours/Week	3	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 03

Course objectives: This course will enable students to

- Explain the fundamental design of the unix operating system
- Familiarize with the systems calls provided in the unix environment
- Design and build an application/service over the unix operating system

Module – 1	**Teaching Hours**
Introduction: UNIX and ANSI Standards: The ANSI C Standard, The ANSI/ISO C++ Standards, Difference between ANSI C and C++, The POSIX Standards, The POSIX.1 FIPS Standard, The X/Open Standards. UNIX and POSIX APIs: The POSIX APIs, The UNIX and POSIX Development Environment, API Common Characteristics.	**8 Hours**
Module – 2	
UNIX Files and APIs: File Types, The UNIX and POSIX File System, The UNIX and POSIX File Attributes, Inodes in UNIX System V, Application Program Interface to Files, UNIX Kernel Support for Files, Relationship of C Stream Pointers and File Descriptors, Directory Files, Hard and Symbolic Links. UNIX File APIs: General File APIs, File and Record Locking, Directory File APIs, Device File APIs, FIFO File APIs, Symbolic Link File APIs.	**8 Hours**
Module – 3	
UNIX Processes and Process Control: The Environment of a UNIX Process: Introduction, main function, Process Termination, Command-Line Arguments, Environment List, Memory Layout of a C Program, Shared Libraries, Memory Allocation, Environment Variables, setjmp and longjmp Functions, getrlimit, setrlimit Functions, UNIX Kernel Support for Processes. Process Control: Introduction, Process Identifiers, fork, vfork, exit, wait, waitpid, wait3, wait4 Functions, Race Conditions, exec Functions, Changing User IDs and Group IDs, Interpreter Files, system Function, Process Accounting, User Identification, Process Times, I/O Redirection. Process Relationships: Introduction, Terminal Logins, Network Logins, Process Groups, Sessions, Controlling Terminal, tcgetpgrp and tcsetpgrp Functions, Job Control, Shell Execution of Programs, Orphaned Process Groups.	**8 Hours**
Module – 4	
Signals and Daemon Processes: Signals: The UNIX Kernel Support for Signals, signal, Signal Mask, sigaction, The SIGCHLD Signal and the waitpid Function, The sigsetjmp and siglongjmp Functions, Kill, Alarm, Interval Timers, POSIX.lb Timers. Daemon Processes: Introduction, Daemon Characteristics, Coding Rules, Error Logging, Client-Server Model.	**8 Hours**
Module – 5	
Interprocess Communication : Overview of IPC Methods, Pipes, popen, pclose Functions, Coprocesses, FIFOs, System V IPC, Message Queues, Semaphores.	**8 Hours**

Shared Memory, Client-Server Properties, Stream Pipes, Passing File Descriptors, An Open Server-Version 1, Client-Server Connection Functions.	

Course outcomes: The students should be able to:

- Ability to understand and reason out the working of Unix Systems
- Build an application/service over a Unix system.

Question paper pattern:

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Unix System Programming Using C++ - Terrence Chan, PHI, 1999.
2. Advanced Programming in the UNIX Environment - W.Richard Stevens, Stephen A. Rago, 3nd Edition, Pearson Education / PHI, 2005.

Reference Books:

1. Advanced Unix Programming- Marc J. Rochkind, 2nd Edition, Pearson Education, 2005.
2. The Design of the UNIX Operating System - Maurice.J.Bach, Pearson Education / PHI, 1987.
3. Unix Internals - Uresh Vahalia, Pearson Education, 2001.

SOFT AND EVOLUTIONARY COMPUTING
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VII

Subject Code	15CS751	IA Marks	20
Number of Lecture Hours/Week	3	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 03

Course objectives: This course will enable students to

- Familiarize with the basic concept of soft computing and intelligent systems
- Compare with various intelligent systems
- Analyze the various soft computing techniques

Module – 1	**Teaching Hours**
Introduction to soft computing: ANN, FS,GA, SI, ES, Comparing among intelligent systems ANN: introduction, biological inspiration, BNN&ANN, classification, first Generation NN, perceptron, illustrative problems **Text Book 1: Chapter1: 1.1-1.8, Chapter2: 2.1-2.6**	**8 Hours**
Module – 2	
Adaline, Medaline, ANN: (2^{nd} generation), introduction, BPN, KNN,HNN, BAM, RBF,SVM and illustrative problems **Text Book 1: Chapter2: 3.1,3.2,3.3,3.6,3.7,3.10,3.11**	**8 Hours**
Module – 3	
Fuzzy logic: introduction, human learning ability, undecidability, probability theory, classical set and fuzzy set, fuzzy set operations, fuzzy relations, fuzzy compositions, natural language and fuzzy interpretations, structure of fuzzy inference system, illustrative problems **Text Book 1: Chapter 5**	**8 Hours**
Module – 4	
Introduction to GA, GA, procedures, working of GA, GA applications, applicability, evolutionary programming, working of EP, GA based Machine learning classifier system, illustrative problems **Text Book 1: Chapter 7**	**8 Hours**
Module – 5	
Swarm Intelligent system: Introduction, Background of SI, Ant colony system Working of ACO, Particle swarm Intelligence(PSO). **Text Book 1: 8.1-8.4, 8.7**	**8 Hours**

Course outcomes: The students should be able to:

- Understand soft computing techniques
- Apply the learned techniques to solve realistic problems
- Differentiate soft computing with hard computing techniques

Question paper pattern:
The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:
1. Soft computing : N. P Padhy and S P Simon , Oxford University Press 2015
Reference Books:
1. Principles of Soft Computing, Shivanandam, Deepa S. N Wiley India, ISBN 13: 2011

COMPUTER VISION AND ROBOTICS
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VII

Subject Code	15CS752	IA Marks	20
Number of Lecture Hours/Week	3	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 03

Course objectives: This course will enable students to

- Review image processing techniques for computer vision
- Explain shape and region analysis
- Illustrate Hough Transform and its applications to detect lines, circles, ellipses
- Contrast three-dimensional image analysis techniques, motion analysis and applications of computer vision algorithms

Module – 1	**Teaching Hours**
CAMERAS: Pinhole Cameras**, Radiometry – Measuring Light:** Light in Space, Light Surfaces, Important Special Cases, **Sources, Shadows, And Shading:** Qualitative Radiometry, Sources and Their Effects, Local Shading Models, Application: Photometric Stereo, Interreflections: Global Shading Models, **Color:** The Physics of Color, Human Color Perception, Representing Color, A Model for Image Color, Surface Color from Image Color.	**8 Hours**
Module – 2	
Linear Filters: Linear Filters and Convolution, Shift Invariant Linear Systems, Spatial Frequency and Fourier Transforms, Sampling and Aliasing, Filters as Templates, **Edge Detection:** Noise, Estimating Derivatives, Detecting Edges, **Texture:** Representing Texture, Analysis (and Synthesis) Using Oriented Pyramids, Application: Synthesis by Sampling Local Models, Shape from Texture.	**8 Hours**
Module – 3	
The Geometry of Multiple Views: Two Views, **Stereopsis:** Reconstruction, Human Stereposis, Binocular Fusion, Using More Cameras, **Segmentation by Clustering:** What Is Segmentation?, Human Vision: Grouping and Getstalt, Applications: Shot Boundary Detection and Background Subtraction, Image Segmentation by Clustering Pixels, Segmentation by Graph-Theoretic Clustering,	**8 Hours**
Module – 4	
Segmentation by Fitting a Model: The Hough Transform, Fitting Lines, Fitting Curves, Fitting as a Probabilistic Inference Problem, Robustness, **Segmentation and Fitting Using Probabilistic Methods:** Missing Data Problems, Fitting, and Segmentation, The EM Algorithm in Practice, **Tracking With Linear Dynamic Models:** Tracking as an Abstract Inference Problem, Linear Dynamic Models, Kalman Filtering, Data Association, Applications and Examples**.**	**8 Hours**
Module – 5	
Geometric Camera Models: Elements of Analytical Euclidean Geometry, Camera Parameters and the Perspective Projection, Affine Cameras and Affine Projection Equations, **Geometric Camera Calibration:** Least-Squares Parameter Estimation, A Linear Approach to Camera Calibration, Taking Radial Distortion into Account, Analytical Photogrammetry, An Application: Mobile Robot Localization, **Model- Based Vision:** Initial Assumptions, Obtaining	**8 Hours**

Hypotheses by Pose Consistency, Obtaining Hypotheses by pose Clustering, Obtaining Hypotheses Using Invariants, Verification, Application: Registration In Medical Imaging Systems, Curved Surfaces and Alignment.	

Course outcomes: The students should be able to:

- Implement fundamental image processing techniques required for computer vision
- Perform shape analysis
- Implement boundary tracking techniques
- Apply chain codes and other region descriptors
- Apply Hough Transform for line, circle, and ellipse detections.
- Apply 3D vision techniques.
- Implement motion related techniques.
- Develop applications using computer vision techniques.

Question paper pattern:

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. David A. Forsyth and Jean Ponce: Computer Vision – A Modern Approach, PHI Learning (Indian Edition), 2009.

Reference Books:

2. E. R. Davies: Computer and Machine Vision – Theory, Algorithms and Practicalities, Elsevier (Academic Press), 4th edition, 2013.

INFORMATION MANAGEMENT SYSTEM
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VII

Subject Code	15IS753	IA Marks	20
Number of Lecture Hours/Week	4	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 03

Course objectives: This course will enable students to

- Explain the Role of information management system in business
- Evaluate the role of the major types of information systems in a business environment and their relationship to each other

Module – 1	**Teaching Hours**
Information Systems in Business : Introduction, The real world of Information Systems, Networks, What you need to know, The fundamental role of IS in business, Trends in IS, Managerial challenges of IT. System Concepts: A foundation, Components of an Information System, Information System Resources, Information System activities, Recognizing Information Systems. Fundamentals of strategic advantages: Strategic IT, Competitive strategy concepts, The competitive advantage of IT, Strategic uses of IT, Building a customer-focused business, The value chain and strategic IS, Reengineering business processes, Becoming an agile company Creating a virtual company, Building a knowledge-creating company.	**08 Hours**
Module – 2	
Enterprise Business Systems: Introduction, Cross-functional enterprise applications, Enterprise application integration, Transaction processing systems, Enterprise collaboration systems. Functional Business Systems: Introduction, Marketing systems, Manufacturing systems, Human resource systems, Accounting systems, Financial management systems.	**08 Hours**
Module – 3	
Customer relationship management: Introduction, What is CRM? The three phases of CRM, Benefits and challenges of CRM, Trends in CRM Enterprise resource planning: Introduction, What is ERP? Benefits and challenges of ERP, Trends in ERP. Supply chain Management: Introduction, What is SCM? The role of SCM, Benefits and challenges of SCM, Trends in SCM.	**08 Hours**
Module – 4	
Electronic commerce fundamentals: Introduction, The scope of ecommerce, Essential e-commerce, processes, Electronic payment processes. e-Commerce applications and issues: E-commerce application trends, Business-to- Consumer e-commerce, Web store requirements, Business-to- Business e-commerce, e-commerce marketplaces, Clicks and bricks in ecommerce	**08 Hours**
Module – 5	
Decision support in business: Introduction, Decision support trends, Decision support systems (DSS), Management Information Systems, Online analytical processing, Using DSS, Executive information systems, Enterprise portals and decision support, Knowledge management systems, Business and Artificial Intelligence (AI), An overview of AI, Expert systems.	**08 Hours**

Course outcomes: The students should be able to:

- Describe the role of information technology and information systems in business
- Record the current issues of information technology and relate those issues to the firm
- Interpret how to use information technology to solve business problems

Question paper pattern:

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. James A.O'Brien, George M Marakas, Management Information Systems, 7th Edition, Tata McGrawHill. Chapter: 1, 2, 7 , 8 ,9 ,13

Reference Books:

2. Kenneth C. Laudon and Jane P.Laudon, Management Information System, Managing the Digital Firm, 9th Edition, Pearson Education.
3. Steven Alter, Information Systems the Foundation of E-Business, 4th Edition, Pearson Education.
4. W.S.Jawadekar, Management Information System, Tata McGraw Hill

STORAGE AREA NETWORKS
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VII

Subject Code	15CS754	IA Marks	20
Number of Lecture Hours/Week	3	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 03

Course objectives: This course will enable students to

- Evaluate storage architectures,
- Define backup, recovery, disaster recovery, business continuity, and replication
- Examine emerging technologies including IP-SAN
- Understand logical and physical components of a storage infrastructure
- Identify components of managing and monitoring the data center
- Define information security and identify different storage virtualization technologies

Module – 1	**Teaching Hours**
Storage System Introduction to evolution of storage architecture, key data center elements, virtualization, and cloud computing. Key data center elements – Host (or compute), connectivity, storage, and application in both classic and virtual environments. RAID implementations, techniques, and levels along with the impact of RAID on application performance.Components of intelligent storage systems and virtual storage provisioning and intelligent storage system implementations.	**8 Hours**
Module – 2	
Storage Networking Technologies and Virtualization Fibre Channel SAN components, connectivity options, and topologies including access protection mechanism ‘zoning”, FC protocol stack, addressing and operations, SAN-based virtualization and VSAN technology, iSCSI and FCIP protocols for storage access over IP network, Converged protocol FCoE and its components, Network Attached Storage (NAS) - components, protocol and operations, File level storage virtualization, Object based storage and unified storage platform.	**8 Hours**
Module – 3	
Backup, Archive, and Replication This unit focuses on information availability and business continuity solutions in both virtualized and non-virtualized environments. Business continuity terminologies, planning and solutions, Clustering and multipathing architecture to avoid single points of failure, Backup and recovery - methods, targets and topologies, Data deduplication and backup in virtualized environment, Fixed content and data archive, Local replication in classic and virtual environments, Remote replication in classic and virtual environments, Three-site remote replication and continuous data protection	**8 Hours**
Module – 4	
Cloud Computing Characteristics and benefits This unit focuses on the business drivers, definition, essential characteristics, and phases of journey to the Cloud. ,Business drivers for Cloud computing, Definition of Cloud computing, Characteristics of Cloud computing, Steps involved in transitioning from Classic data center to Cloud computing environment Services and deployment models, Cloud infrastructure components, Cloud migration considerations	**8 Hours**
Module – 5	

Securing and Managing Storage Infrastructure This chapter focuses on framework and domains of storage security along with covering security. implementation at storage networking. Security threats, and countermeasures in various domains Security solutions for FC-SAN, IP-SAN and NAS environments, Security in virtualized and cloud environments, Monitoring and managing various information infrastructure components in classic and virtual environments, Information lifecycle management (ILM) and storage tiering, Cloud service management activities	**8 Hours**

Course outcomes: The students should be able to:

- Identify key challenges in managing information and analyze different storage networking technologies and virtualization
- Explain components and the implementation of NAS
- Describe CAS architecture and types of archives and forms of virtualization
- Ilustrate the storage infrastructure and management activities

Question paper pattern:

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Information Storage and Management,Author :EMC Education Services, Publisher: Wiley ISBN: 9781118094839
2. Storage Virtualization, Author: Clark Tom, Publisher: Addison Wesley Publishing Company ISBN : 9780321262516

Reference Books:

NIL

MACHINE LEARNING LABORATORY
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VII

Subject Code	15CSL76	IA Marks	20
Number of Lecture Hours/Week	01I + 02P	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 02

Course objectives: This course will enable students to

1. Make use of Data sets in implementing the machine learning algorithms
2. Implement the machine learning concepts and algorithms in any suitable language of choice.

Description (If any):

1. The programs can be implemented in either JAVA or Python.
2. For Problems 1 to 6 and 10, programs are to be developed without using the built-in classes or APIs of Java/Python.
3. Data sets can be taken from standard repositories (https://archive.ics.uci.edu/ml/datasets.html) or constructed by the students.

Lab Experiments:

1. Implement and demonstratethe **FIND-Salgorithm** for finding the most specific hypothesis based on a given set of training data samples. Read the training data from a .CSV file.
2. For a given set of training data examples stored in a .CSV file, implement and demonstrate the **Candidate-Elimination algorithm**to output a description of the set of all hypotheses consistent with the training examples.
3. Write a program to demonstrate the working of the decision tree based **ID3 algorithm**. Use an appropriate data set for building the decision tree and apply this knowledge toclassify a new sample.
4. Build an Artificial Neural Network by implementing the **Backpropagation algorithm** and test the same using appropriate data sets.
5. Write a program to implement the **naïve Bayesian classifier** for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.
6. Assuming a set of documents that need to be classified, use the **naïve Bayesian Classifier** model to perform this task. Built-in Java classes/API can be used to write the program. Calculate the accuracy, precision, and recall for your data set.
7. Write a program to construct a**Bayesian network** considering medical data. Use this model to demonstrate the diagnosis of heart patients using standard Heart Disease Data Set. You can use Java/Python ML library classes/API.
8. Apply **EM algorithm** to cluster a set of data stored in a .CSV file. Use the same data set for clustering using ***k*-Means algorithm**. Compare the results of these two algorithms and comment on the quality of clustering. You can add Java/Python ML library classes/API in the program.
9. Write a program to implement ***k*-Nearest Neighbour algorithm** to classify the iris data set. Print both correct and wrong predictions. Java/Python ML library classes can be used for this problem.
10. Implement the non-parametric **Locally Weighted Regressionalgorithm** in order to fit data points. Select appropriate data set for your experiment and draw graphs.

Study Experiment / Project:
NIL
Course outcomes: The students should be able to:
1. Understand the implementation procedures for the machine learning algorithms. 2. Design Java/Python programs for various Learning algorithms. 3. Applyappropriate data sets to the Machine Learning algorithms. 4. Identify and apply Machine Learning algorithms to solve real world problems.
Conduction of Practical Examination:
• All laboratory experiments are to be included for practical examination. • Students are allowed to pick one experiment from the lot. • Strictly follow the instructions as printed on the cover page of answer script • Marks distribution: Procedure + Conduction + Viva:**20 + 50 +10 (80)** **Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.**

WEB TECHNOLOGY LABORATORY WITH MINI PROJECT
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VII

Subject Code	15CSL77	IA Marks	20
Number of Lecture Hours/Week	01I + 02P	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 02

Course objectives: This course will enable students to

1. Design and develop static and dynamic web pages.
2. Familiarize with Client-Side Programming, Server-Side Programming, Active server Pages.
3. Learn Database Connectivity to web applications.

Description (If any):

NIL

Lab Experiments:

PART A

1. Write a JavaScript to design a simple calculator to perform the following operations: sum, product, difference and quotient.
2. Write a JavaScript that calculates the squares and cubes of the numbers from 0 to 10 and outputs HTML text that displays the resulting values in an HTML table format.
3. Write a JavaScript code that displays text "TEXT-GROWING" with increasing font size in the interval of 100ms in RED COLOR, when the font size reaches 50pt it displays "TEXT-SHRINKING" in BLUE color. Then the font size decreases to 5pt.
4. Develop and demonstrate a HTML5 file that includes JavaScript script that uses functions for the following problems:
   a. Parameter: A string
   b. Output: The position in the string of the left-most vowel
   c. Parameter: A number
   d. Output: The number with its digits in the reverse order
5. Design an XML document to store information about a student in an engineering college affiliated to VTU. The information must include USN, Name, and Name of the College, Branch, Year of Joining, and email id. Make up sample data for 3 students. Create a CSS style sheet and use it to display the document.
6. Write a PHP program to keep track of the number of visitors visiting the web page and to display this count of visitors, with proper headings.
7. Write a PHP program to display a digital clock which displays the current time of the server.
8. Write the PHP programs to do the following:
   a. Implement simple calculator operations.
   b. Find the transpose of a matrix.
   c. Multiplication of two matrices.
   d. Addition of two matrices.

9. Write a PHP program named states.py that declares a variable states with value "Mississippi Alabama Texas Massachusetts Kansas". write a PHP program that does the following:
    a. Search for a word in variable states that ends in xas. Store this word in element 0 of a list named statesList.
    b. Search for a word in states that begins with k and ends in s. Perform a case-insensitive comparison. [Note: Passing re.Ias a second parameter to method compile performs a case-insensitive comparison.] Store this word in element1 of statesList.
    c. Search for a word in states that begins with M and ends in s. Store this word in element 2 of the list.
    d. Search for a word in states that ends in a. Store this word in element 3 of the list.
10. Write a PHP program to sort the student records which are stored in the database using selection sort.

Study Experiment / Project:

Develop a web application project using the languages and concepts learnt in the theory and exercises listed in part A with a good look and feel effects. You can use any web technologies and frameworks and databases.

Note:

1. In the examination each student picks one question from part A.

2. A team of two or three students must develop the mini project. However during the examination, each student must demonstrate the project individually.

3. The team must submit a brief project report (15-20 pages) that must include the following
    a. Introduction
    b. Requirement Analysis
    c. Software Requirement Specification
    d. Analysis and Design
    e. Implementation
    f. Testing

Course outcomes: The students should be able to:

- Design and develop dynamic web pages with good aesthetic sense of designing and latest technical know-how's.
- Have a good understanding of Web Application Terminologies, Internet Tools other web services.
- Learn how to link and publish web sites

Conduction of Practical Examination:

1. All laboratory experiments from part A are to be included for practical examination.

2. Mini project has to be evaluated for 30 Marks.
3. Report should be prepared in a standard format prescribed for project work.
4. Students are allowed to pick one experiment from the lot.
5. Strictly follow the instructions as printed on the cover page of answer script.
6. Marks distribution:
    a) Part A: Procedure + Conduction + Viva:10 + 35 +5 =50 Marks
    b) Part B: Demonstration + Report + Viva voce = 15+10+05 = 30 Marks

Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.

INTERNET OF THINGS TECHNOLOGY
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VIII

Subject Code	**15CS81**	IA Marks	20
Number of Lecture Hours/Week	04	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03

CREDITS – 04

Course Objectives: This course will enable students to

- Assess the genesis and impact of IoT applications, architectures in real world.
- Illustrate diverse methods of deploying smart objects and connect them to network.
- Compare different Application protocols for IoT.
- Infer the role of Data Analytics and Security in IoT.
- Identifysensor technologies for sensing real world entities and understand the role of IoT in various domains of Industry.

Module – 1	**Teaching Hours**
What is IoT, Genesis of IoT, IoT and Digitization, IoT Impact, Convergence of IT and IoT, IoT Challenges, IoT Network Architecture and Design, Drivers Behind New Network Architectures, Comparing IoT Architectures, A Simplified IoT Architecture, The Core IoT Functional Stack, IoT Data Management and Compute Stack.	**10 Hours**
Module – 2	
Smart Objects: The "Things" in IoT, Sensors, Actuators, and Smart Objects, Sensor Networks, Connecting Smart Objects, Communications Criteria, IoT Access Technologies.	**10 Hours**
Module – 3	
IP as the IoT Network Layer, The Business Case for IP, The need for Optimization, Optimizing IP for IoT, Profiles and Compliances, Application Protocols for IoT, The Transport Layer, IoT Application Transport Methods.	**10 Hours**
Module – 4	
Data and Analytics for IoT, An Introduction to Data Analytics for IoT, Machine Learning, Big Data Analytics Tools and Technology, Edge Streaming Analytics, Network Analytics, Securing IoT, A Brief History of OT Security, Common Challenges in OT Security, How IT and OT Security Practices and Systems Vary, Formal Risk Analysis Structures: OCTAVE and FAIR, The Phased Application of Security in an Operational Environment	**10 Hours**
Module – 5	
IoT Physical Devices and Endpoints - Arduino UNO: Introduction to Arduino, Arduino UNO, Installing the Software, Fundamentals of Arduino Programming. IoT Physical Devices and Endpoints - RaspberryPi: Introduction to RaspberryPi, About the RaspberryPi Board: Hardware Layout, Operating Systems on RaspberryPi, Configuring RaspberryPi, Programming RaspberryPi with Python, Wireless Temperature Monitoring System Using Pi, DS18B20 Temperature Sensor, Connecting Raspberry Pi via SSH, Accessing Temperature from DS18B20 sensors, Remote access to RaspberryPi, Smart and Connected Cities, An IoT Strategy for Smarter Cities, Smart City IoT Architecture,	**10 Hours**

Smart City Security Architecture, Smart City Use-Case Examples.	

Course Outcomes: After studying this course, students will be able to

- Interpret the impact and challenges posed by IoT networks leading to new architectural models.
- Compare and contrast the deployment of smart objects and the technologies to connect them to network.
- Appraise the role of IoT protocols for efficient network communication.
- Elaborate the need for Data Analytics and Security in IoT.
- Illustrate different sensor technologies for sensing real world entities and identify the applications of IoT in Industry.

Question paper pattern:

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Robert Barton, Jerome Henry,**"IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things",** 1st Edition, Pearson Education (Cisco Press Indian Reprint). (**ISBN:** 978-9386873743)
2. Srinivasa K G, **"Internet of Things",** CENGAGE Leaning India, 2017

Reference Books:

1. Vijay Madisetti and ArshdeepBahga, **"Internet of Things (A Hands-on-Approach)",** 1st Edition, VPT, 2014. (**ISBN:** 978-8173719547)
2. Raj Kamal, **"Internet of Things: Architecture and Design Principles",** 1st Edition, McGraw Hill Education, 2017. (**ISBN:** 978-9352605224)

BIG DATA ANALYTICS
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VIII

Subject Code	15CS82	IA Marks	20
Number of Lecture Hours/Week	4	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03

CREDITS – 04

Course objectives: This course will enable students to

- Understand Hadoop Distributed File system and examine MapReduce Programming
- Explore Hadoop tools and manage Hadoop with Ambari
- Appraise the role of Business intelligence and its applications across industries
- Assess core data mining techniques for data analytics
- Identify various Text Mining techniques

Module – 1	**Teaching Hours**
Hadoop Distributed File System Basics, Running Example Programs and Benchmarks, Hadoop MapReduce Framework, MapReduce Programming	**10 Hours**
Module – 2	
Essential Hadoop Tools, Hadoop YARN Applications, Managing Hadoop with Apache Ambari, Basic Hadoop Administration Procedures	**10 Hours**
Module – 3	
Business Intelligence Concepts and Application, Data Warehousing, Data Mining, Data Visualization	**10 Hours**
Module – 4	
Decision Trees, Regression, Artificial Neural Networks, Cluster Analysis, Association Rule Mining	**10 Hours**
Module – 5	
Text Mining, Naïve-Bayes Analysis, Support Vector Machines, Web Mining, Social Network Analysis	**10 Hours**

Course outcomes: The students should be able to:

- Master the concepts of HDFS and MapReduce framework
- Investigate Hadoop related tools for Big Data Analytics and perform basic Hadoop Administration
- Recognize the role of Business Intelligence, Data warehousing and Visualization in decision making
- Infer the importance of core data mining techniques for data analytics
- Compare and contrast different Text Mining Techniques

Question paper pattern:
The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Douglas Eadline,**"Hadoop 2 Quick-Start Guide: Learn the Essentials of Big Data Computing in the Apache Hadoop 2 Ecosystem",** 1st Edition, Pearson Education, 2016. ISBN-13: 978-9332570351

2. Anil Maheshwari, **“Data Analytics”,** 1st Edition, McGraw Hill Education, 2017. ISBN-13: 978-9352604180

Reference Books:

1) Tom White, **“Hadoop: The Definitive Guide”,** 4th Edition, O’Reilly Media, 2015.ISBN-13: 978-9352130672
2) Boris Lublinsky, Kevin T.Smith, Alexey Yakubovich,**"Professional Hadoop Solutions",** 1stEdition, Wrox Press, 2014ISBN-13: 978-8126551071
3) Eric Sammer,**"Hadoop Operations: A Guide for Developers and Administrators",**1stEdition, O'Reilly Media, 2012.ISBN-13: 978-9350239261

HIGH PERFORMANCE COMPUTING
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VIII

Subject Code	15CS831	IA Marks	20
Number of Lecture Hours/Week	3	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 03

Course objectives: This course will enable students to

- Introduce students the design, analysis, and implementation, of high performance computational science and engineering applications.
- Illustrate on advanced computer architectures, parallel algorithms, parallel languages, and performance-oriented computing.

Module – 1	**Teaching Hours**
Introduction: Computational Science and Engineering: Computational Science and Engineering Applications; characteristics and requirements, Review of Computational Complexity, Performance: metrics and measurements, Granularity and Partitioning, Locality: temporal/spatial/stream/kernel, Basic methods for parallel programming, Real-world case studies (drawn from multi-scale, multi-discipline applications)	**10 Hours**
Module – 2	
High-End Computer Systems : Memory Hierarchies, Multi-core Processors: Homogeneous and Heterogeneous, Shared-memory Symmetric Multiprocessors, Vector Computers, Distributed Memory Computers, Supercomputers and Petascale Systems, Application Accelerators / Reconfigurable Computing, Novel computers: Stream, multithreaded, and purpose-built	**10 Hours**
Module – 3	
Parallel Algorithms: Parallel models: ideal and real frameworks, Basic Techniques: Balanced Trees, Pointer Jumping, Divide and Conquer, Partitioning, Regular Algorithms: Matrix operations and Linear Algebra, Irregular Algorithms: Lists, Trees, Graphs, Randomization: Parallel Pseudo-Random Number Generators, Sorting, Monte Carlo techniques	**10 Hours**
Module – 4	
Parallel Programming: Revealing concurrency in applications, Task and Functional Parallelism, Task Scheduling, Synchronization Methods, Parallel Primitives (collective operations), SPMD Programming (threads, OpenMP, MPI), I/O and File Systems, Parallel Matlabs (Parallel Matlab, Star-P, Matlab MPI), Partitioning Global Address Space (PGAS) languages (UPC, Titanium, Global Arrays)	**10 Hours**
Module – 5	
Achieving Performance: Measuring performance, Identifying performance bottlenecks, Restructuring applications for deep memory hierarchies, Partitioning applications for heterogeneous resources, using existing libraries, tools, and frameworks	**10 Hours**

Course outcomes: The students should be able to:

- Illustrate the key factors affecting performance of CSE applications, and
- Make mapping of applications to high-performance computing systems, and

- Apply hardware/software co-design for achieving performance on real-world applications

Question paper pattern:

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Introduction to Parallel Computing, AnanthGrama, Anshul Gupta, George Karypis, and Vipin Kumar, 2nd edition, Addison-Welsey, 2003.
2. Petascale Computing: Algorithms and Applications, David A. Bader (Ed.), Chapman & Hall/CRC Computational Science Series, 2007

Reference Books:

1. Grama, A. Gupta, G. Karypis, V. Kumar, An Introduction to Parallel Computing, Design and Analysis of Algorithms: 2/e, Addison-Wesley, 2003.
2. G.E. Karniadakis, R.M. Kirby II, Parallel Scientific Computing in C++ and MPI: A Seamless Approach to Parallel Algorithms and their Implementation, Cambridge University Press,2003.
3. Wilkinson and M. Allen, Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers, 2/E, Prentice Hall, 2005.
4. M.J. Quinn, Parallel Programming in C with MPI and OpenMP, McGraw-Hill, 2004.
5. G.S. Almasi and A. Gottlieb, Highly Parallel Computing, 2/E, Addison-Wesley, 1994.
6. David Culler Jaswinder Pal Singh,"Parallel Computer Architecture: A hardware/Software Approach", Morgan Kaufmann, 1999.
7. Kai Hwang, "Scalable Parallel Computing", McGraw Hill 1998.

USER INTERFACE DESIGN
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VIII

Subject Code	15CS832	IA Marks	20
Number of Lecture Hours/Week	3	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 03

Course objectives: This course will enable students to

- To study the concept of menus, windows, interfaces
- To study about business functions
- To study the characteristics and components of windows andthe various controls for the windows.
- To study about various problems in windows design with color, text, graphics.
- To study the testing methods

Module – 1	**Teaching Hours**
Introduction-Importance-Human-Computer interface-characteristics of graphics interface-Direct manipulation graphical system - web user interface-popularity-characteristic & principles.	**10 Hours**
Module – 2	
User interface design process- obstacles-usability-human characteristics in design - Human interaction speed-business functions-requirement analysis-Direct-Indirect methods-basic business functions-Design standards-system timings - Human consideration in screen design - structures of menus - functions of menus-contents of menu-formatting -phrasing the menu - selecting menu choice-navigating menus-graphical menus.	**10 Hours**
Module – 3	
Windows: Characteristics-components-presentation styles-types-managements-organizations-operations-web systems-device-based controls: characteristics-Screen -based controls: operate control - text boxes-selection control-combination control-custom control-presentation control.	**10 Hours**
Module – 4	
Text for web pages - effective feedback-guidance & assistance-Internationalization-accessibility -Icons-Image-Multimedia-coloring.	**10 Hours**
Module – 5	
Windows layout-test :prototypes - kinds of tests - retest - Information search - visualization - Hypermedia - www - Software tools.	**10 Hours**

Course outcomes: The students should be able to:

- Design the user interface, design, menu creation and windows creation and connection between menu and windows

Question paper pattern:
The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Wilbent. O. Galitz ,"The Essential Guide to User Interface Design", John Wiley&

Sons, 2001.

Reference Books:

1. Ben Sheiderman, "Design the User Interface", Pearson Education, 1998.
2. Alan Cooper, "The Essential of User Interface Design", Wiley - Dream Tech Ltd., 2002.

VIRTUAL REALITY
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VIII

Subject Code	15IS833	IA Marks	20
Number of Lecture Hours/Week	3	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 03

Course objectives: This course will enable students to

- Explain understanding of this technology, underlying principles, its potential and limits and to learn about the criteria for defining useful applications.
- Illustrate process of creating virtual environments

Module – 1	**Teaching Hours**
Introduction : The three I's of virtual reality, commercial VR technology and the five classic components of a VR system. Input Devices : (Trackers, Navigation, and Gesture Interfaces): Three-dimensional position trackers, navigation and manipulation, interfaces and gesture interfaces. **Text book1: 1.1, 1.3, 1.5, 2.1, 2.2 and 2.3**	**10 Hours**
Module – 2	
Output Devices: Graphics displays, sound displays & haptic feedback. **Text book1: 3.1,3.2 and 3.3**	**10 Hours**
Module – 3	
Modeling : Geometric modeling, kinematics modeling, physical modeling, behaviour modeling, model management. **Text book1: 5.1, 5.2 and 5.3, 5.4 and 5.5**	**10 Hours**
Module – 4	
Human Factors: Methodology and terminology, user performance studies, VR health and safety issues. **Text book1: 7.1, 7.2 and 7.3**	**10 Hours**
Module – 5	
Applications: Medical applications, military applications, robotics applications. **Text book1: 8.1, 8.3 and 9.2**	**10 Hours**

Course outcomes: The students should be able to:

- Illustrate technology, underlying principles, its potential and limits and to learn about the criteria for defining useful applications.
- Explain process of creating virtual environments

Question paper pattern:
The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Virtual Reality Technology, Second Edition, Gregory C. Burdea & Philippe Coiffet, John Wiley & Sons

Reference Books:

SYSTEM MODELLING AND SIMULATION
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VIII

Subject Code	15CS834	IA Marks	20
Number of Lecture Hours/Week	3	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 03

Course objectives: This course will enable students to

- Explain the basic system concept and definitions of system;
- Discuss techniques to model and to simulate various systems;
- Analyze a system and to make use of the information to improve the performance.

Module – 1	**Teaching Hours**
Introduction: When simulation is the appropriate tool and when it is not appropriate, Advantages and disadvantages of Simulation; Areas of application, Systems and system environment; Components of a system; Discrete and continuous systems, Model of a system; Types of Models, Discrete-Event System Simulation Simulation examples: Simulation of queuing systems. **General Principles, Simulation Software:**Concepts in Discrete-Event Simulation. The Event-Scheduling / Time-Advance Algorithm, Manual simulation Using Event Scheduling	**10 Hours**
Module – 2	
Statistical Models in Simulation :Review of terminology and concepts, Useful statistical models,Discrete distributions. Continuous distributions,Poisson process, Empirical distributions. **Queuing Models:**Characteristics of queuing systems,Queuing notation,Long-run measures of performance of queuing systems,Long-run measures of performance of queuing systems cont…,Steady-state behavior of M/G/1 queue, Networks of queues,	**10 Hours**
Module – 3	
Random-NumberGeneration: Properties of random numbers; Generation of pseudo-random numbers, Techniques for generating random numbers,Tests for Random Numbers, **Random-Variate Generation:** ,Inverse transform technique Acceptance-Rejection technique.	**10 Hours**
Module – 4	
Input Modeling: Data Collection; Identifying the distribution with data, Parameter estimation, Goodness of Fit Tests, Fitting a non-stationary Poisson process, Selecting input models without data, Multivariate and Time-Series input models. **Estimation of Absolute Performance:** Types of simulations with respect to output analysis ,Stochastic nature of output data, Measures of performance and their estimation, **Contd..**	**10 Hours**
Module – 5	
Measures of performance and their estimation,Output analysis for terminating simulations Continued..,Output analysis for steady-state simulations. **Verification, Calibration And Validation:** Optimization: Model building, verification and validation, Verification of simulation models, Verification of	**10 Hours**

simulation models,Calibration and validation of models, Optimization via Simulation.	

Course outcomes: The students should be able to:

- Explain the system concept and apply functional modeling method to model the activities of a static system
- Describe the behavior of a dynamic system and create an analogous model for a dynamic system;
- Simulate the operation of a dynamic system and make improvement according to the simulation results.

Question paper pattern:

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Jerry Banks, John S. Carson II, Barry L. Nelson, David M. Nicol: Discrete-Event System Simulation, 5 th Edition, Pearson Education, 2010.

Reference Books:

1. Lawrence M. Leemis, Stephen K. Park: Discrete – Event Simulation: A First Course, Pearson Education, 2006.
2. Averill M. Law: Simulation Modeling and Analysis, 4 th Edition, Tata McGraw-Hill, 2007

INTERNSHIP / PROFESSIONAL PRACTISE [As per Choice Based Credit System (CBCS) scheme] (Effective from the academic year 2016 -2017) SEMESTER – VIII			
Subject Code	15CS84	IA Marks	50
Duration	4 weeks	Exam Marks	50
		Exam Hours	03
CREDITS – 02			
Course objectives: This course will enable students to			
Description (If any):			
Course outcomes: The students should be able to:			
Evaluation of Internship :			

PROJECT WORK PHASE II [As per Choice Based Credit System (CBCS) scheme] (Effective from the academic year 2016 -2017) SEMESTER – VIII			
Subject Code	15CSP85	IA Marks	100
Number of Lecture Hours/Week	06	Exam Marks	100
Total Number of Lecture Hours	--	Exam Hours	03
CREDITS – 05			
Course objectives: This course will enable students to			
Description (If any):			
Course outcomes: The students should be able to:			
Conduction of Practical Examination:			

SEMINAR [As per Choice Based Credit System (CBCS) scheme] (Effective from the academic year 2016 -2017) SEMESTER – VIII			
Subject Code	15CSS86	IA Marks	100
Number of Lecture Hours/Week	04	Exam Marks	--
Total Number of Lecture Hours	--	Exam Hours	--
CREDITS – 02			
Course objectives: This course will enable students to			
•			
Description:			
•			
Course outcomes: The students should be able to:			
•			
Evaluation of seminar:			