Learning Outcome based Curriculum Framework (LOCF)

For

Choice Based Credit System (CBCS)

Syllabus

B.Sc. (Honours) in Computer Science

w.e.f. Academic Session 2020-21

Kazi Nazrul University

Asansol, Paschim Bardhaman

West Bengal 713340

Preamble

The objective of any programme at a Higher Education Institution is to create for its students a sound foundation for their character development which directly contributes to the well-being of a nation. Kazi Nazrul University envisions all its programmes in the spirit of its "motto" which is to inspire the youth to show steadfastness and devotion in a fearless pursuit of truth. The LOCF aims at preparing young minds for constructive and productive character development by honing their creative and humanistic skills for their own betterment as well as for the greater good of the society. In order to provide an opportunity to students to discover a method of thinking which will help them realise their true potential, the University offers a Learning Outcome-based Curriculum Framework (LOCF) for all its Under Graduate programmes.

The LOCF approach is intended to provide focused, outcome-based syllabi at the undergraduate level with an agenda to structure the teaching-learning experiences in a more student-centric manner by making the courses flexible and by offering students more choices. The LOCF approach has been adopted to strengthen the teacher- learner interaction as students engage themselves in programmes of their choice and learn to realize their inner calling. As the Under- Graduate Programmes focus on 'preparing minds', they will create individuals who will have intellectual prowess, interactive competence, courage to lead the world and also compassion and empathy for fellow human beings. The LOCF thus aims at strengthening not merely students' employability skills but also at imparting to them vital life-skills required to lead a happy personal and social life.

Each programme vividly elaborates its nature and promises the outcomes that are to be accomplished by studying the courses. The programmes also state the attributes that they offer to inculcate at the graduation level. The graduate attributes encompass values related to students' well-being, emotional stability, critical thinking etc. intermingled with a sense of social justice and harmony. In short, each programme prepares students for employability, sustainability and life-long learning. The new curriculum will empower students to innovate and also inspire them to convert their innovations into real business models for the country's economic and social prosperity. The proposed LOCF offers better understanding of the business world and aims at building students' entrepreneurial skills by giving them hands-on training. The Kazi Nazrul University hopes the LOCF approach of the programme will motivate students to transition from being passive knowledge-seekers to becoming active and aware knowledge-creators.

Semester- I

Course Name: PROGRAMMING METHODOLOGY

Course Code: BSCHCOSC101

<table>
<tr><td colspan="2">Course Type: Core (Theoretical & Practical)</td><td colspan="2">Course Details: CC-1</td><td colspan="2">L-T-P: 4 - 0 - 4</td></tr>
<tr><td rowspan="3">Credit: 6</td><td rowspan="3">Full Marks: 100</td><td colspan="2">CA Marks</td><td colspan="2">ESE Marks</td></tr>
<tr><td>Practical</td><td>Theoretical</td><td>Practical</td><td>Theoretical</td></tr>
<tr><td>30</td><td>10</td><td>20</td><td>40</td></tr>
</table>

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Learn to develop simple algorithms and flow charts to solve a problem.*
2. *Develop problem solving skills coupled with top down design principles.*
3. *Learn about the strategies of writing efficient and well-structured computer algorithms/programs.*
4. *Develop the skills for formulating iterative solutions to a problem.*
5. *Learn array processing algorithms coupled with iterative methods.*
6. *Learn text and string processing efficient algorithms.*
7. *Learn searching techniques and use of pointers.*
8. *Understand recursive techniques in programming.*

Course Content:

Theory

UNIT I. Introduction to Programming, Program Concept, Characteristics of Programming, Stages in Program Development, Algorithms, Notations, Design, Flowcharts, Types of Programming Methodologies, Introduction to C/C++ Programming - Basic Program Structure In C/C++, Variables and Assignments, Input and Output, Selection and Repetition Statements.

UNIT II. Top-Down Design, Predefined Functions, Programmer -defined Function, Local Variable, Functions with Default Arguments

UNIT III. Introduction to Arrays, Declaration and Referring Arrays, Arrays in Memory, Initializing Arrays. Arrays in Functions, Multi-Dimensional Arrays.

UNIT IV. Pointers - Understanding a Pointer Variable, Simple use of Pointers (Declaring and Dereferencing Pointers to simple variables), Pointers to Pointers, Call-By-Value and Call-By-Reference Parameters.

UNIT V. Structures - Member Accessing, Pointers to Structures, Structures and Functions, Arrays of Structures, Unions.

UNIT VI. Strings - Declaration and Initialization, Reading and Writing Strings, Arrays of Strings, String and Function, Strings and Structure, Standard String Library Functions.

UNIT VII. Searching Algorithms - Linear Search, Binary Search. Use of files for data input and output. merging and copy files.

UNIT VIII. Recursion - Developing Recursive Definition of Simple Problems and their implementation.

Practical

UNIT I. Given the problem statement, students are required to formulate problem, develop flowchart/algorithm, write code, execute and test it. Students should be given assignments on following :

a) To learn elementary techniques involving arithmetic operators and mathematical expressions, appropriate use of selection (if, switch, conditional operators) and control structures

b) Learn how to use functions and parameter passing in functions, writing recursive programs.

UNIT II. Students should be given assignments on following:

a) Write Programs to learn the use of strings and string handling operations.

b) Problems which can effectively demonstrate use of Arrays. Structures and Union.

c) Write programs using pointers.

d) Write programs to use files for data input and output.

e) Write programs to implement search algorithms.

Internal (CA) Evaluation: Practical Note Book (15 marks), Two experiments (10 marks) – one from each unit, Viva-voce (5 marks)

ESE Evaluation: Two experiments (10 marks) – one from each unit, Viva-voce (10 marks)

References/ Suggested Readings:

1. Problem Solving and Program Design in C, J. R. Hanly and E. B. Koffman, Pearson, 2015.
2. Programming and problem solving with C++: brief edition, N. Dale and C. Weems, Jones & Bartlett Learning, 2010.
3. C Programming, Karnighan,&Ritchie, PHI
4. Programming through C, Richard Johnsonbaugh and Martin Kalin, Pearson Education
5. Programming in C, B.S. Gottfried, Sahaum Series.
6. Programming in ANSI C, E. Balaguruswami, TMH

Course Name: Computer System Architecture

Course Code: BSCHCOSC102

<table>
<tr><td rowspan="2">Course Type: Core (Theoretical & Practical)</td><td colspan="3" rowspan="2">Course Details: CC-2</td><td colspan="2" rowspan="2">L-T-P: 4 - 0 - 4</td></tr>
<tr></tr>
<tr><td rowspan="3">Credit: 6</td><td rowspan="3">Full Marks: 100</td><td colspan="2">CA Marks</td><td colspan="2">ESE Marks</td></tr>
<tr><td>Practical</td><td>Theoretical</td><td>Practical</td><td>Theoretical</td></tr>
<tr><td>30</td><td>10</td><td>20</td><td>40</td></tr>
</table>

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. To make students understand the basic structure, operation and characteristics of digital computer.
2. To familiarize the students with arithmetic and logic unit as well as the concept of the concept of pipelining.
3. To familiarize the students with hierarchical memory system including cache memories and virtual memory.
4. To make students know the different ways of communicating with I/O devices and standard I/O interfaces.

Course Content:

Theory

UNIT I. Introduction - Logic gates (OR, AND, NOT, NAND, NOR, Exclusive – OR, Exclusive – NOR, Mixed logic), Boolean algebra, Map Simplification (Sum of Products, Product of Sums, Karnaugh Map (up to 4 variables)), combinational circuits (Adder, Subtractor, Multiplexer, Demultiplexer, Decoder, Encoder, Comparator, Code converter), Circuit simplification, flip-flops (SR,T,D,JK,MASTER SLAVE) and sequential circuits (registers, counters and memory units).

UNIT II. Data Representation and Basic Computer Arithmetic - Number systems, complements, fixed and floating point representation, character representation, addition, subtraction, magnitude comparison, multiplication and division algorithms for integers.

UNIT III. Register Transfer and Micro operations: Register Transfer Language, Register Transfer, Bus & Memory Transfer, Arithmetic Microoperations, Logic Microoperations, Shift Microoperation.

UNIT IV. Basic Computer Organization: Instruction codes, Computer Registers, Computer Instructions, Timing & Control, Instruction Cycles, Memory Reference Instruction, Input - Output & Interrupts, Complete Computer Description & Design of Basic Computer.

UNIT V. Processor and Control Unit: Hardwired vs. Micro programmed Control Unit, General Register Organization, Stack Organization, Instruction Format, Data Transfer & Manipulation, Program Control, RISC, CISC, Pipelining – Pipelined datapath and control – Handling Data hazards & Control hazards. Introduction to Parallelism.

UNIT VI. Memory and I/O Systems: Peripheral Devices, I/O Interface, Data Transfer Schemes, Program Control, Interrupt, DMA Transfer, I/O Processor. Memory Hierarchy, Processor vs. Memory Speed, High-Speed Memories, Cache Memory, Associative Memory, Interleave, Virtual Memory, Cache Mapping Techniques, Memory Management.

Practical

UNIT I. Design of combinational circuits.

UNIT II. Design of sequential circuits.

UNIT III. Create a fetch routine of the instruction cycle.

UNIT IV. Create a machine based on the given architecture (Register Sets, Memory, Instruction format and basic computer instructions).

Internal (CA) Evaluation: Practical Note Book (15 marks), One experiment (10 marks) – from Unit III or Unit IV, Viva-voce (5 marks)

ESE Evaluation: Two experiments (10 marks) – one from Unit I and another from Unit II, Viva-voce (10 marks)

References/ Suggested Readings:

1. Digital Logic and Computer Design, M. Morris Mao, PHI.
2. Computer System Architecture, M. Morris Mano, 3rd Edition, Prentice Hall.
3. Computer Organization and Design, David A. Patterson and John L. Hennessey, Fifth edition, Morgan Kauffman / Elsevier, 2014.
4. Floyd, Digital Fundamentals, Pearson Education.
5. Computer Architecture: A Quantitative Approach, John L. Hennessy, David A. Patterson, 4th Edition.
6. Computer Organization and Architecture, William Stallings, Prentice Hall.

Course Name: PROGRAMMING METHODOLOGY

Course Code: BSCHCOSGE101

<table>
<tr><td colspan="2">Course Type: GE (Theoretical & Practical)</td><td colspan="2">Course Details: GEC-1</td><td colspan="2">L-T-P: 4 - 0 - 4</td></tr>
<tr><td rowspan="3">Credit: 6</td><td rowspan="3">Full Marks: 100</td><td colspan="2">CA Marks</td><td colspan="2">ESE Marks</td></tr>
<tr><td>Practical</td><td>Theoretical</td><td>Practical</td><td>Theoretical</td></tr>
<tr><td>30</td><td>10</td><td>20</td><td>40</td></tr>
</table>

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Learn to develop simple algorithms and flow charts to solve a problem.*
2. *Develop problem solving skills coupled with top down design principles.*
3. *Learn about the strategies of writing efficient and well-structured computer algorithms/programs.*
4. *Develop the skills for formulating iterative solutions to a problem.*
5. *Learn array processing algorithms coupled with iterative methods.*
6. *Learn text and string processing efficient algorithms.*
7. *Learn searching techniques and use of pointers.*
8. *Understand recursive techniques in programming.*

Course Content:

Theory

UNIT I. Introduction to Programming, Program Concept, Characteristics of Programming, Stages in Program Development, Algorithms, Notations, Design, Flowcharts, Types of Programming Methodologies, Introduction to C/C++ Programming - Basic Program Structure In C/C++, Variables and Assignments, Input and Output, Selection and Repetition Statements.

UNIT II. Top-Down Design, Predefined Functions, Programmer -defined Function, Local Variable, Functions with Default Arguments

UNIT III. Introduction to Arrays, Declaration and Referring Arrays, Arrays in Memory, Initializing Arrays. Arrays in Functions, Multi-Dimensional Arrays.

UNIT IV. Pointers - Understanding a Pointer Variable, Simple use of Pointers (Declaring and Dereferencing Pointers to simple variables), Pointers to Pointers, Call-By-Value and Call-By-Reference Parameters.

UNIT V. Structures - Member Accessing, Pointers to Structures, Structures and Functions, Arrays of Structures, Unions.

UNIT VI. Strings - Declaration and Initialization, Reading and Writing Strings, Arrays of Strings, String and Function, Strings and Structure, Standard String Library Functions.

UNIT VII. Searching Algorithms - Linear Search, Binary Search. Use of files for data input and output. merging and copy files.

UNIT VIII. Recursion - Developing Recursive Definition of Simple Problems and their implementation.

Practical

UNIT I. Given the problem statement, students are required to formulate problem, develop flowchart/algorithm, write code, execute and test it. Students should be given assignments on following:

a. To learn elementary techniques involving arithmetic operators and mathematical expressions, appropriate use of selection (if, switch, conditional operators) and control structures

b. Learn how to use functions and parameter passing in functions, writing recursive programs.

UNIT II. Students should be given assignments on following:

a. Write Programs to learn the use of strings and string handling operations.

b. Problems which can effectively demonstrate use of Arrays. Structures and Union.

c. Write programs using pointers.

d. Write programs to use files for data input and output.

e. Write programs to implement search algorithms.

Internal (CA) Evaluation: Practical Note Book (15 marks), Two experiments (10 marks) – one from each unit, Viva-voce (5 marks)

ESE Evaluation: Two experiments (10 marks) – one from each unit, Viva-voce (10 marks)

References/ Suggested Readings:

1. Problem Solving and Program Design in C, J. R. Hanly and E. B. Koffman, Pearson, 2015.
2. Programming and problem solving with C++: brief edition, N. Dale and C. Weems, Jones & Bartlett Learning, 2010.
3. C Programming, Karnighan,&Ritchie, PHI
4. Programming through C, Richard Johnsonbaugh and Martin Kalin, Pearson Education
5. Programming in C, B.S. Gottfried, Sahaum Series.
6. Programming in ANSI C, E. Balaguruswami, TMH

Semester- II

Course Name: DATA STRUCTURE

Course Code: BSCHCOSC201

<table>
<tr><td>Course Type: Core (Theoretical & Practical)</td><td colspan="2">Course Details: CC-3</td><td colspan="2">L-T-P: 4 - 0 - 4</td></tr>
<tr><td rowspan="3">Credit: 6</td><td rowspan="3">Full Marks: 100</td><td colspan="2">CA Marks</td><td colspan="2">ESE Marks</td></tr>
<tr><td>Practical</td><td>Theoretical</td><td>Practical</td><td>Theoretical</td></tr>
<tr><td>30</td><td>10</td><td>20</td><td>40</td></tr>
</table>

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *To be familiar with fundamental data structures and with the manner in which these data structures can best be implemented; become accustomed to the description of algorithms in both functional and procedural styles*
2. *To have knowledge of complexity of basic operations like insert, delete, search on these data structures.*
3. *Ability to choose a data structure to suitably model any data used in computer applications.*
4. *Design programs using various data structures including hash tables, Binary and general search trees, heaps, graphs etc.*
5. *Ability to assess efficiency tradeoffs among different data structure implementations.*
6. *Implement and know the applications of algorithms for sorting, searching etc.*

Course Content:

Theory

UNIT I. Basic concepts- Algorithm Specification-Introduction, Recursive algorithms, Data Abstraction, Performance analysis, Linear and Non Linear data structures, Singly Linked Lists-Operations, Concatenating, Circularly linked lists-Operations for Circularly linked lists, Doubly Linked Lists- Operations. Representation of single, two dimensional arrays, sparse matrices-array and linked representations.

UNIT II. Stack- Definition and Operations, Array and Linked Implementations, Applications - Valid Expression Checking (Parenthesis matching), Reversal of string, Infix to Postfix Conversion, Postfix Expression Evaluation, Recursion Implementation.

UNIT III. Queue - Definition and Operations, Array and Linked Implementations, Applications, Circular Queues - Insertion and Deletion Operations, Dequeue (Double Ended Queue) - Introduction.

UNIT IV. Sorting Methods – Bubble, Insertion, Selection, Shell, Using Divide-Conquer Approach (Quick and Merge sort), Comparison of Sorting Methods, Searching Methods – Linear and Binary.

UNIT V. Trees, Representation of Trees, Binary tree, Properties of Binary Trees, Binary Tree Representations- Array and Linked Representations, Binary Tree Traversals, Threaded Binary Trees, Binary Search tree - Creation, Insertion, Deletion and Search, AVL tree- Definition, Examples, Insertion and Rotations, B tree, B+ tree, Priority Queue- Definition and Implementation, Heap- Definition, Min heap, Max heap, Insertion and Deletion.

UNIT VI. Static Hashing- Introduction, Hash tables, Hash functions, Overflow Handling.

Practical

Students are required to write and practically execute programs to solve problem using various data structures. The teacher can suitably device problems which help students experiment using the suitable data structures and operations. Some of the problems are indicated below.

1. Write program that uses functions to perform the following:
   a) Creation of list of elements where the size of the list, elements to be inserted and deleted are dynamically given as input.
   b) Implement the operations, insertion, deletion at a given position in the list and search for an element in the list
   c) To display the elements in forward / reverse order
2. Write a program to implement stack data structure and basic operations on it (Insertion, deletion). Write a program that demonstrates the application of stack operations (Eg: infix expression to postfix conversion, postfix evaluation).
3. Write a program to implement queue data structure and basic operations on it (Insertion, deletion, find length) and code at least one application using queues.
4. Write program that implements linear and binary search methods of searching for an elements in a list.
5. Write and trace programs to understand the various phases of sorting elements using the methods a) Bubble sort b) Insertion Sort c) Quicksort etc.
6. Write a program to create a Binary search tree and insert and delete from the tree. Write recursive and non-recursive routines to traverse a binary tree in preorder, inorder and postorder.
7. Write programs for recursion (Eg. Fibonacci numbers, Towers of Hanoi).

Internal (CA) Evaluation: Practical Note Book (15 marks), Two experiments (10 marks), Viva-voce (5 marks)

ESE Evaluation: Two experiments (10 marks), Viva-voce (10 marks)

References/ Suggested Readings:

1. Fundamentals of Data structures in C, 2nd Edition, E. Horowitz, S. Sahni and Susan Anderson-Freed, Universities Press.

2. Data structures and Algorithm Analysis in C, 2nd edition, M. A. Weiss, Pearson.

3. Lipschutz: Schaum's outline series Data structures Tata McGraw-Hill

4. Data Structure through C in Depth. S.K. Srivastava and Deepali Srivastava, B.P.B Publication.

Course Name: DISCRETE STRUCTURES

Course Code: BSCHCOSC202

Course Type: **Core (Theoretical)**	Course Details: **CC-4**			L-T-P: **5 - 1 - 0**	
Credit: **6**	Full Marks: **50**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		….	**10**	….	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Understand the notion of mathematical thinking, mathematical proofs, and algorithmic thinking, and be able to apply them in problem solving.*
2. *Understand the basics of combinatories, and be able to apply the methods from these subjects in problem solving.*
3. *Be able to use effectively algebraic techniques to analyse basic discrete structures and algorithms.*
4. *Understand asymptotic notation, its significance, and be able to use it to analyse asymptotic performance for some basic algorithmic examples.*
5. *Understand some basic properties of graphs and related discrete structures, and be able to relate these to practical examples.*

Course Content:

Theory

UNIT I. **Sets**: Finite and Infinite Sets, Uncountable Infinite Sets; **Functions:** Domain, Co-domain, Range, Equal function, Exponential function, Logarithmic function, Square function, Cube function, Relations: Reflexive, Symmetric, Anti-symmetric, Properties of Binary

Relations, Closure, Partial Ordering Relations; Counting - Pigeonhole Principle, **Permutation and Combination**: Introduction to Permutation and Combination, Permutation of thing not all different, Multiplication Principle, Addition Principle; **Mathematical Induction**, Principle of Inclusion and Exclusion.

UNIT II. Growth of Functions: Asymptotic Notations, Summation Formulas and Properties, Bounding Summations, Approximation by Integrals.

UNIT III. Recurrences: Recurrence Relations, Generating Functions, Linear Recurrence Relations with Constant Coefficients and their Solution, Substitution Method, Recurrence Trees, Master Theorem.

UNIT IV. Graph Theory: Basic Terminology, Models and Types, Multigraphs and Weighted Graphs, Directed Graph, Graph Representation, Graph Isomorphism, Connectivity, Euler and Hamiltonian Paths and Circuits, Planar Graphs, Graph Coloring, Trees, Basic Terminology and Properties of Trees, Introduction to Spanning Trees.

UNIT V. Propositional Logic: Proposition Or Statements, Truth table, Logical Connectives, Well-formed Formulas, Tautologies, Contradiction, Equivalences, Inference Theory, Conjunctive Normal Form, Disjunctive Normal Form.

References/ Suggested Readings:

1. C.L. Liu & Mahopatra, Elements of Discrete mathematics, 2nd Sub Edition 1985, Tata McGraw Hill.

2. Kenneth Rosen, Discrete Mathematics and Its Applications, Sixth Edition, McGraw Hill 2006

3. M. O. Albertson and J. P. Hutchinson, Discrete Mathematics with Algorithms 1988 John wiley Publication.

Course Name: DATA STRUCTURE

Course Code: BSCHCOSGE201

<table>
<tr><td rowspan="2">Course Type: GE (Theoretical & Practical)</td><td colspan="2" rowspan="2">Course Details: GEC-2</td><td colspan="3">L-T-P: 4 - 0 - 4</td></tr>
<tr></tr>
<tr><td rowspan="3">Credit: 6</td><td rowspan="3">Full Marks: 100</td><td colspan="2">CA Marks</td><td colspan="2">ESE Marks</td></tr>
<tr><td>Practical</td><td>Theoretical</td><td>Practical</td><td>Theoretical</td></tr>
<tr><td>30</td><td>10</td><td>20</td><td>40</td></tr>
</table>

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *To be familiar with fundamental data structures and with the manner in which these data structures can best be implemented; become accustomed to the description of algorithms in both functional and procedural styles*
2. *To have knowledge of complexity of basic operations like insert, delete, search on these data structures.*
3. *Ability to choose a data structure to suitably model any data used in computer applications.*
4. *Design programs using various data structures including hash tables, Binary and general search trees, heaps, graphs etc.*
5. *Ability to assess efficiency tradeoffs among different data structure implementations.*
6. *Implement and know the applications of algorithms for sorting, searching etc.*

Course Content:

Theory

UNIT I. Basic concepts- Algorithm Specification-Introduction, Recursive algorithms, Data Abstraction, Performance analysis, Linear and Non Linear data structures, Singly Linked Lists-Operations, Concatenating, Circularly linked lists-Operations for Circularly linked lists, Doubly Linked Lists- Operations. Representation of single, two dimensional arrays, sparse matrices-array and linked representations.

UNIT II. Stack- Definition and Operations, Array and Linked Implementations, Applications - Valid Expression Checking (Parenthesis matching), Reversal of string, Infix to Postfix Conversion, Postfix Expression Evaluation, Recursion Implementation.

UNIT III. Queue - Definition and Operations, Array and Linked Implementations, Applications, Circular Queues - Insertion and Deletion Operations, Dequeue (Double Ended Queue) - Introduction.

UNIT IV. Sorting Methods – Bubble, Insertion, Selection, Shell, Using Divide-Conquer Approach (Quick and Merge sort), Comparison of Sorting Methods, Searching Methods – Linear and Binary.

UNIT V. Trees, Representation of Trees, Binary tree, Properties of Binary Trees, Binary Tree Representations- Array and Linked Representations, Binary Tree Traversals, Threaded Binary Trees, Binary Search tree - Creation, Insertion, Deletion and Search, AVL tree- Definition, Examples, Insertion and Rotations, B tree, B+ tree, Priority Queue- Definition and Implementation, Heap- Definition, Min heap, Max heap, Insertion and Deletion.

UNIT VI. Graphs, Graph ADT, Graph Representations, Graph Traversals and Searching, Static Hashing- Introduction, Hash tables, Hash functions, Overflow Handling.

Practical

Students are required to write and practically execute programs to solve problem using various data structures. The teacher can suitably device problems which help students experiment using the suitable data structures and operations. Some of the problems are indicated below.

1. Write program that uses functions to perform the following:
   a) Creation of list of elements where the size of the list, elements to be inserted and deleted are dynamically given as input.
   b) Implement the operations, insertion, deletion at a given position in the list and search for an element in the list
   c) To display the elements in forward / reverse order
2. Write a program to implement stack data structure and basic operations on it (Insertion, deletion). Write a program that demonstrates the application of stack operations (Eg: infix expression to postfix conversion, postfix evaluation).
3. Write a program to implement queue data structure and basic operations on it (Insertion, deletion, find length) and code at least one application using queues.
4. Write program that implements linear and binary search methods of searching for an elements in a list.
5. Write and trace programs to understand the various phases of sorting elements using the methods a) Bubble sort b) Insertion Sort c) Quicksort etc.
6. Write a program to create a Binary search tree and insert and delete from the tree. Write recursive and non-recursive routines to traverse a binary tree in preorder, inorder and postorder.
7. Write programs for recursion (Eg. Fibonacci numbers, Towers of Hanoi).
8. Represent suitably a graph data structure and demonstrate operations of traversals on it.

Internal (CA) Evaluation: Practical Note Book (15 marks), Two experiments (10 marks), Viva-voce (5 marks)

ESE Evaluation: Two experiments (10 marks), Viva-voce (10 marks)

References/ Suggested Readings:

1. Fundamentals of Data structures in C, 2nd Edition, E. Horowitz, S. Sahni and Susan Anderson-Freed, Universities Press.
2. Data structures and Algorithm Analysis in C, 2nd edition, M. A. Weiss, Pearson.
3. Lipschutz: Schaum's outline series Data structures Tata McGraw-Hill
4. Data Structure through C in Depth. S.K. Srivastava and Deepali Srivastava, B.P.B Publication.

Semester- III

Course Name: Operating System

Course Code: BSCHCOSC301

<table>
<tr><td colspan="2">Course Type: Core (Theoretical & Practical)</td><td colspan="2">Course Details: CC-5</td><td colspan="2">L-T-P: 4 - 0 - 4</td></tr>
<tr><td rowspan="3">Credit: 6</td><td rowspan="3">Full Marks: 100</td><td colspan="2">CA Marks</td><td colspan="2">ESE Marks</td></tr>
<tr><td>Practical</td><td>Theoretical</td><td>Practical</td><td>Theoretical</td></tr>
<tr><td>30</td><td>10</td><td>20</td><td>40</td></tr>
</table>

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Describe the important computer system resources and the role of operating system in their management policies and algorithms.*
2. *To understand various functions, structures and history of operating systems and should be able to specify objectives of modern operating systems and describe how operating systems have evolved over time.*
3. *Understanding of design issues associated with operating systems.*
4. *Understand various process management concepts including scheduling, synchronization, and deadlocks.*
5. *To have a basic knowledge about multithreading.*
6. *To understand concepts of memory management including virtual memory.*
7. *To understand issues related to file system interface and implementation, disk management.*
8. *To understand and identify potential threats to operating systems and the security features design to guard against them.*
9. *To have sound knowledge of various types of operating systems including Unix and Android.*
10. *Describe the functions of a contemporary operating system with respect to convenience, efficiency, and the ability to evolve.*

Course Content:

Theory

UNIT I. (Introduction to Operating System) What is Operating System? History and Evolution of OS, Basic OS functions, Resource Abstraction, Types of Operating Systems– Multiprogramming Systems, Batch Systems, Time Sharing Systems; Operating Systems for Personal Computers, Workstations and Hand-held Devices, Process Control & Real time Systems.

UNIT II. (Operating System Organization and Process Characterization) Processor and User Modes, Kernels, System Calls and System Programs, System View of the Process and Resources, Process Abstraction, Process Hierarchy, Threads, Threading Issues, Thread Libraries; Process Scheduling, Non-Pre-emptive and Preemptive Scheduling Algorithms.

UNIT III. Process Management (Deadlock) Deadlock, Deadlock Characterization, Necessary and Sufficient Conditions for Deadlock, Deadlock Handling Approaches: Deadlock Prevention, Deadlock Avoidance and Deadlock Detection and Recovery.

UNIT IV. (Inter Process Communication and Synchronization) Concurrent and Dependent Processes, Critical Section, Semaphores, Methods for Inter-process Communication; Process Synchronization, Classical Process Synchronization Problems: Producer-Consumer, Reader-Writer.

UNIT V. (Memory Management) Physical and Virtual Address Space; Memory Allocation Strategies– Fixed and -Variable Partitions, Paging, Segmentation, Virtual Memory; Page Replacement Algorithms.

UNIT VI. (File and I/O Management, Disk Scheduling, OS security) Directory Structure, File Operations, File Allocation Methods, Device Management, Pipes, Buffer, Shared Memory, Disk Scheduling (FCFS, SSTF, SCAN, C-SCAN, LOOK, C-LOOK), Security Policy Mechanism, Protection, Authentication and Internal Access Authorization.

UNIT VII. (Introduction to Android Operating System) Introduction to Android Operating System, Android Development Framework, Android Application Architecture, Android Process Management and File System.

Practical

UNIT I. Students are required to write and practically execute programs to solve following problems using C programming language.

1. WRITE A PROGRAM (using fork() and/or exec() commands) where parent and child execute: a) same program, same code. b) same program, different code. c) before terminating, the parent waits for the child to finish its task.
2. WRITE A PROGRAM to report behavior of Linux kernel including kernel version, CPU type and model. (CPU information)
3. WRITE A PROGRAM to report behavior of Linux kernel including information on configured memory, amount of free and used memory. (memory information)
4. WRITE A PROGRAM to print file details including owner access permissions, file access time, where file name is given as argument.
5. WRITE A PROGRAM to copy files using system calls.
6. Write programs to implement scheduling algorithms (FCFS, Round Robin, SJF, SRJF)
7. Write program to implement non-preemptive priority based scheduling algorithm.
8. Write program to implement preemptive priority based scheduling algorithm.
9. Write program to calculate sum of n numbers using thread library.
10. Write a program to implement first-fit, best-fit and worst-fit allocation strategies.

UNIT II. UNIX and Shell Scripts

1. External and internal commands of UNIX
2. What is shell and various type of shell, Various editors present in unix/linux
3. Different modes of operation in vi editor
4. What is shell script, Writing and executing the shell script
5. Shell variable (user defined and system variables)
6. System calls, Using system calls
7. Pipes and Filters
8. Decision making in Shell Scripts (If else, switch), Loops in shell
9. Functions
10. Utility programs (cut, paste, join, tr, uniq utilities), Pattern matching utility (grep).

Internal (CA) Evaluation: Practical Note Book (15 marks), Two experiments (10 marks) – one from each unit, Viva-voce (5 marks)

ESE Evaluation: Two experiments (10 marks) – one from each unit, Viva-voce (10 marks)

References/ Suggested Readings:

1. A Silberschatz, P.B. Galvin, G. Gagne, Operating Systems Concepts, 8th Edition, John Wiley Publications 2008.
2. A.S. Tanenbaum, Modern Operating Systems, 3rd Edition, Pearson Education 2007.
3. G. Nutt, Operating Systems: A Modern Perspective, 2nd Edition Pearson Education 1997.
4. W. Stallings, Operating Systems, Internals & Design Principles 2008 5th Edition, Prentice Hall of India. M. Milenkovic, Operating Systems- Concepts and design, Tata McGraw Hill 1992.
5. Sumitabha, Das, Unix Concepts and Applications, Tata McGraw-Hill Education.
6. Nemeth Synder and Hein, Linux Administration Handbook, Pearson Education, 2nd Edition ,2010.
7. W. Richard Stevens, Bill Fenner, Andrew M. Rudoff, Unix Network Programming, The sockets Networking API, Vol. 1, 3rd Edition, 2014.
8. Yashavant Kanetkar , UNIX Shell Programming, BPB Publication.
9. Kernighan and Pike, The Unix Programming Environment, Prentice-Hall.

Course Name: Analysis of Algorithms

Course Code: BSCHCOSC302

<table>
<tr><td>Course Type:
Core (Theoretical)</td><td colspan="2">Course Details: CC-6</td><td colspan="2">L-T-P: 5 - 1 - 0</td></tr>
<tr><td rowspan="3">Credit: 6</td><td rowspan="3">Full Marks:
50</td><td colspan="2">CA Marks</td><td colspan="2">ESE Marks</td></tr>
<tr><td>Practical</td><td>Theoretical</td><td>Practical</td><td>Theoretical</td></tr>
<tr><td>….</td><td>10</td><td>….</td><td>40</td></tr>
</table>

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *To learn good principles of algorithm design;*
2. *To learn how to analyse algorithms and estimate their worst-case and average case behaviour (in easy cases);*
3. *To become familiar with fundamental data structures and with the manner in which these data structures can best be implemented; become accustomed to the description of algorithms in both functional and procedural styles;*

Course Content:

Theory

UNIT I. Introduction: Basic Design and Analysis Techniques of Algorithms, Correctness of Algorithm. Growth of Functions: Asymptotic notation, Big-O, Theta, Omega notations. Algorithm Design Techniques: Iterative Techniques, Divide and Conquer, Dynamic Programming, Greedy Algorithms.

UNIT II. Greedy algorithm: Characteristics and features of problem solving by greedy algorithm, basic structure, feasibility, Huffman code, Dijkstra. Knapsack problem (0/1, fractional)

UNIT III. Sorting and Searching Techniques: Elementary Sorting techniques– Bubble Sort, Insertion Sort, Merge Sort, Advanced Sorting techniques- Heap Sort, Quick Sort, Sorting in Linear Time - Bucket Sort, Radix Sort and Count Sort, Searching Techniques- Medians & Order Statistics, complexity analysis

UNIT IV. Graphs Algorithms: Graph Algorithms– Breadth First Search, Depth First Search and its Applications, Spanning tree, Minimum Spanning Trees (Kruskal and Prim's algorithms). String Processing (KMP Technique)

UNIT V. Lower Bounding Techniques: Decision Trees, Balanced Trees (AVL,B tree, Red-Black Trees)

UNIT VI. Advanced Analysis Technique: Randomized Algorithm, Distributed Algorithm, Heuristics

References/ Suggested Readings:

1. T.H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein Introduction to Algorithms, PHI, 3rd Edition 2009
2. Sara basse & A.V. Gelder, Computer Algorithm – Introduction to Design and Analysis, Publisher – Pearson 3rd Edition 1999
3. Knuth Donald E., The art of computer programming: Fundamental algorithms (Vol. 1), Pearson. 3rd Edition.

Course Name: Computer Networks

Course Code: BSCHCOSC303

Course Type: **Core (Theoretical & Practical)**	Course Details: **CC-7**			L-T-P: **4 - 0 - 4**	
Credit: **6**	Full Marks: **100**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		30	**10**	**20**	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Understand the structure of Data Communications System and its components. Be familiarize with different network terminologies.*
2. *Familiarize with contemporary issues in network technologies.*
3. *Know the layered model approach explained in OSI and TCP/IP network models*
4. *Identify different types of network devices and their functions within a network.*
5. *Learn basic routing mechanisms, IP addressing scheme and internetworking concepts.*
6. *Familiarize with IP and TCP Internet protocols.*
7. *To understand major concepts involved in design of WAN, LAN and wireless networks.*
8. *Learn basics of network configuration and maintenance.*
9. *Know the fundamentals of network security issues.*

Course Content:

Theory

UNIT I. Introduction to Computer Networks and Networking Elements: Network Definition, Network Topologies, Network Classifications, Network Protocol, Layered Network Architecture, Overview of OSI Reference Model, Overview of TCP/IP Protocol Suite, Hub, Switch (Managed and Unmanaged), Routers.

UNIT II. Data Communication Fundamentals and Techniques: Analog and Digital Signal, Data-Rate Limits, Digital to Digital Line Encoding Schemes, Pulse Code Modulation, Parallel and Serial Transmission, Digital to Analog Modulation - Multiplexing Techniques-FDM, TDM, Transmission Media.

UNIT III. Networks Switching Techniques and Access Mechanisms: Circuit Switching, Packet Switching- Connectionless Datagram Switching, ConnectionOriented Virtual Circuit Switching; Dial-Up Modems, Digital Subscriber Line, Cable TV for Data Transfer.

UNIT IV. Data Link Layer Functions and Protocol: Error Detection and Error Correction Techniques, Data-Link Control- Framing and Flow Control, Error Recovery Protocols-Stop and Wait ARQ, Go-Back-N ARQ, Point to Point Protocol on Internet.

UNIT V. Multiple Access Protocol and Network Layer: CSMA/CD Protocols, Ethernet LANS; Connecting LAN and Back-Bone Networks- Repeaters, Hubs, Switches, Bridges, Router and Gateways, Networks Layer Functions and Protocols, Routing, Routing Algorithms, Network Layer Protocol of Internet - IP Protocol, Internet Control Protocols.

UNIT VI. Transport Layer and Application Layer Functions and Protocols: Transport Services- Error and Flow Control, Connection Establishment and Release- Three Way Handshake, Overview of Application Layer Protocol, Overview of DNS Protocol; Overview of WWW & HTTP Protocol.

UNIT VII. Security: Firewall**,** Basics of cryptography; message security; digital signature.

Practical

UNIT I. Network Devices and Configuration

1. Identification of network devices like hub, switch, modem etc.
2. Use of ping and tracert / traceroute, ipconfig / ifconfig, route and arp utilities.
3. Configure LAN
4. Configure IP static routing.
5. Configure IP routing using RIP.

UNIT II. All programs should be developed in C/ C++ / Java / Python

1. Simulate Cyclic Redundancy Check (CRC) error detection algorithm for noisy channel.
2. Simulate Hamming-code based error detection & correction algorithm for noisy channel.
3. Simulate and implement stop and wait protocol for noisy channel.
4. Simulate and implement go back N sliding window protocol.
5. Simulate and implement selective repeat sliding window protocol.
6. Simulate and implement MST construction (Prim‟s, Kruskal‟s) for Ethernet
7. Simulate and implement the various routing algorithms (RIP, Distance-Vector routing, Dijkstra‟s, Bellman-Ford, Floyd-Warshall, Flooding)
8. Socket Programming .

Internal (CA) Evaluation: Practical Note Book (15 marks), One experiments from Unit I (10 marks), Viva-voce (5 marks)

ESE Evaluation: One experiment from Unit II (10 marks), Viva-voce (10 marks)

References/ Suggested Readings:

1. B. A. Forouzan: Data Communications and Networking, Fourth edition, THM Publishing Company Ltd 2007.
2. A. S. Tanenbaum: Computer Networks, Fifth edition, PHI Pvt. Ltd 2011
3. William Stallings:Data and Computer Communications, Eight Edition, Pearson.

Course Name: Operating System and Shell Scripts

Course Code: BSCHCOSGE301

Course Type: **GE (Theoretical & Practical)**	Course Details: **GEC-3**			L-T-P: **4 - 0 - 4**	
Credit: **6**	Full Marks: **100**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		30	**10**	**20**	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Describe the important computer system resources and the role of operating system in their management policies and algorithms.*
2. *To understand various functions, structures and history of operating systems and should be able to specify objectives of modern operating systems and describe how operating systems have evolved over time.*
3. *Understanding of design issues associated with operating systems.*
4. *Understand various process management concepts including scheduling, synchronization, and deadlocks.*
5. *To have a basic knowledge about multithreading.*
6. *To understand concepts of memory management including virtual memory.*
7. *To understand issues related to file system interface and implementation, disk management.*
8. *To understand and identify potential threats to operating systems.*
9. *To have sound knowledge of various types of operating systems including Unix.*
10. *Describe the functions of a contemporary operating system with respect to convenience, efficiency, and the ability to evolve.*

Course Content:

Theory

UNIT I. (Introduction to Operating System) What is Operating System? History and Evolution of OS, Basic OS functions, Resource Abstraction, Types of Operating Systems–

Multiprogramming Systems, Batch Systems, Time Sharing Systems; Operating Systems for Personal Computers, Workstations and Hand-held Devices, Process Control & Real time Systems.

UNIT II. (Operating System Organization and Process Characterization) Processor and User Modes, Kernels, System Calls and System Programs, System View of the Process and Resources, Process Abstraction, Process Hierarchy, Threads, Threading Issues, Thread Libraries; Process Scheduling, Non-Pre-emptive and Preemptive Scheduling Algorithms.

UNIT III. Process Management (Deadlock) Deadlock, Deadlock Characterization, Necessary and Sufficient Conditions for Deadlock, Deadlock Handling Approaches: Deadlock Prevention, Deadlock Avoidance and Deadlock Detection and Recovery.

UNIT IV. (Inter Process Communication and Synchronization) Concurrent and Dependent Processes, Critical Section, Semaphores, Methods for Inter-process Communication; Process Synchronization, Classical Process Synchronization Problems: Producer-Consumer, Reader-Writer.

UNIT V. (Memory Management) Physical and Virtual Address Space; Memory Allocation Strategies– Fixed and -Variable Partitions, Paging, Segmentation, Virtual Memory.

UNIT VI. (File and I/O Management, Disk Scheduling, OS security) Directory Structure, File Operations, File Allocation Methods, Device Management, Pipes, Buffer, Shared Memory, Disk Scheduling (FCFS, SSTF, SCAN, C-SCAN, LOOK, C-LOOK), Security Policy Mechanism, Protection, Authentication and Internal Access Authorization

Practical

UNIT I. External and internal commands of UNIX/Linux
UNIT II. What is shell and various type of shell, Various editors present in Unix/Linux
UNIT III. Different modes of operation in vi editor
UNIT IV. What is shell script, Writing and executing the shell script
UNIT V. Shell variable (user defined and system variables)
UNIT VI. System calls, Using system calls
UNIT VII. Pipes and Filters
UNIT VIII. Decision making in Shell Scripts (If else, switch), Loops in shell
UNIT IX. Functions
UNIT X. Utility programs (cut, paste, join, tr, uniq utilities), Pattern matching utility (grep).

Internal (CA) Evaluation: Practical Note Book (15 marks), Two experiments (10 marks), Viva-voce (5 marks)

ESE Evaluation: Two experiments (10 marks), Viva-voce (10 marks)

References/ Suggested Readings:

1. A Silberschatz, P.B. Galvin, G. Gagne, Operating Systems Concepts, 8th Edition, John Wiley Publications 2008.

2. A.S. Tanenbaum, Modern Operating Systems, 3rd Edition, Pearson Education 2007.
3. W. Stallings, Operating Systems, Internals & Design Principles 2008 5th Edition, Prentice Hall of India. M. Milenkovic, Operating Systems- Concepts and design, Tata McGraw Hill 1992.
4. Sumitabha, Das, Unix Concepts and Applications, Tata McGraw-Hill Education.
5. Yashavant Kanetkar , UNIX Shell Programming, BPB Publication.
6. Kernighan and Pike, The Unix Programming Environment, Prentice-Hall.

Course Name: Introduction to Computer Networks

Course Code: BSCHCOSGE302

<table>
<tr><td rowspan="2">Course Type: GE (Theoretical & Practical)</td><td colspan="3" rowspan="2">Course Details: GEC-3</td><td colspan="2" rowspan="2">L-T-P: 4 - 0 - 4</td></tr>
<tr></tr>
<tr><td rowspan="3">Credit: 6</td><td rowspan="3">Full Marks: 100</td><td colspan="2">CA Marks</td><td colspan="2">ESE Marks</td></tr>
<tr><td>Practical</td><td>Theoretical</td><td>Practical</td><td>Theoretical</td></tr>
<tr><td>30</td><td>10</td><td>20</td><td>40</td></tr>
</table>

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Understand the structure of Data Communications System and its components. Be familiarize with different network terminologies.*
2. *Familiarize with contemporary issues in network technologies.*
3. *Know the layered model approach explained in OSI and TCP/IP network models*
4. *Identify different types of network devices and their functions within a network.*
5. *Learn basic routing mechanisms, IP addressing scheme and internetworking concepts.*
6. *Familiarize with IP and TCP Internet protocols.*
7. *To understand major concepts involved in design of WAN, LAN and wireless networks.*
8. *Learn basics of network configuration and maintenance.*

Course Content:

Theory

UNIT I. Introduction to Computer Networks and Networking Elements: Network Definition, Network Topologies, Network Classifications, Network Protocol, Layered Network Architecture, Overview of OSI Reference Model, Overview of TCP/IP Protocol Suite, Hub, Switch (Managed and Unmanaged), Routers.

UNIT II. Data Communication Fundamentals and Techniques: Analog and Digital Signal, Data-Rate Limits, Digital to Digital Line Encoding Schemes, Pulse Code Modulation,

Parallel and Serial Transmission, Digital to Analog Modulation - Multiplexing Techniques-FDM, TDM, Transmission Media.

UNIT III. Networks Switching Techniques and Access Mechanisms: Circuit Switching, Packet Switching- Connectionless Datagram Switching, ConnectionOriented Virtual Circuit Switching; Dial-Up Modems, Digital Subscriber Line, Cable TV for Data Transfer.

UNIT IV. Data Link Layer Functions and Protocol: Error Detection and Error Correction Techniques, Data-Link Control- Framing and Flow Control, Error Recovery Protocols-Stop and Wait ARQ, Go-Back-N ARQ, Point to Point Protocol on Internet.

UNIT V. Multiple Access Protocol and Network Layer: CSMA/CD Protocols, Ethernet LANS; Connecting LAN and Back-Bone Networks- Repeaters, Hubs, Switches, Bridges, Router and Gateways, Networks Layer Functions and Protocols Routing, Routing Algorithms, Network Layer Protocol of Internet - IP Protocol, Internet Control Protocols.

UNIT VI. Transport Layer and Application Layer Functions and Protocols: Transport Services- Error and Flow Control, Connection Establishment and Release- Three Way Handshake, Overview of Application Layer Protocol, Overview of DNS Protocol; Overview of WWW & HTTP Protocol.

Practical

UNIT I. Network Devices and Configuration

1. Identification of network devices like hub, switch, modem etc.
2. Use of ping and tracert / traceroute, ipconfig / ifconfig, route and arp utilities.
3. Configure LAN
4. Configure IP static routing.
5. Configure IP routing using RIP.

UNIT II. All programs should be developed in C/ C++

1. Simulate Cyclic Redundancy Check (CRC) error detection algorithm for noisy channel.
2. Simulate Hamming-code based error detection & correction algorithm for noisy channel.
3. Simulate and implement stop and wait protocol for noisy channel.
4. Simulate and implement go back N sliding window protocol.
5. Simulate and implement selective repeat sliding window protocol.

Internal (CA) Evaluation: Practical Note Book (15 marks), One experiments from Unit I (10 marks), Viva-voce (5 marks)

ESE Evaluation: One experiment from Unit II (10 marks), Viva-voce (10 marks)

References/ Suggested Readings:

1. B. A. Forouzan: Data Communications and Networking, Fourth edition, THM Publishing Company Ltd 2007.
2. A. S. Tanenbaum: Computer Networks, Fifth edition, PHI Pvt. Ltd 2011.
3. William Stallings:Data and Computer Communications, Eight Edition, Pearson.

Course Name: Programming in Java

Course Code: BSCHCOSSE301

Course Type: **SE (Practical)**	Course Details: **SEC-1**			L-T-P: **0 - 0 - 8**	
Credit: **4**	Full Marks: **50**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		30	**-----**	**20**	**----**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Knowledge of the structure and model of the Java programming language,*
2. *Use the Java programming language for various programming technologies*
3. *Develop software in the Java programming language,*
4. *Evaluate user requirements for software functionality required to decide whether the Java programming language can meet user requirements.*

Course Content:

Practical

UNIT I. Introduction: Java Essentials, Its characteristics, Execution and Compilation, Data types, Variables, Control Statements, Standard Input/ Output.

UNIT II. Constructors, Object Oriented Concepts: Encapsulation, Abstraction, Inheritance, Polymorphisms, JAVA Packages.

UNIT III. Exception Handling, Wrapper Classes, Autoboxing, Multi-thread Programming.

UNIT IV. Applets, Event Handling, AWT.

Students are required to implement object-oriented paradigm using JAVA. Below is the list of some of the experiments.

Part A

1. Program on strings: Check the equality of two strings, Reverse a string.

2. Program using loops: to find the sum of digits of a given number, display a multiplication table, display all prime numbers between 1 to 1000.

3. Program to demonstrate all math class functions.

Part B

4. Program on files: to copy a file to another file using Java to package classes.

5. Program to demonstrate method over-riding and overloading

6. Programs on inheritances.

7. Multi-threaded programming.

Internal (CA) Evaluation: Practical Note Book (15 marks), Two experiments (10 marks) – one from Part A and another from Part B, Viva-voce (5 marks)

ESE Evaluation: Two experiments (10 marks) – one from Part A and another from Part B, Viva-voce (10 marks)

References/ Suggested Readings:

1. E. Balaguruswamy, Programming with Java, 4th Edition, McGraw Hill.2009.
2. John R. Hubbard,"Programming with JAVA, Schaum's Series, 2nd Edition, 2004.
3. Herbert Schildt, The Complete Reference Java 2, TMH.
4. Y. Daniel Liang, Introduction to Java Programming and Data Structures, Pearson, 12th Edition, 2020.

Course Name: Python Programming

Course Code: BSCHCOSSE302

<table>
<tr><td>Course Type:
SE (Practical)</td><td colspan="2">Course Details: SEC-1</td><td colspan="2">L-T-P: 0 - 0 - 8</td></tr>
<tr><td rowspan="3">Credit: 4</td><td rowspan="3">Full Marks:
50</td><td colspan="2">CA Marks</td><td colspan="2">ESE Marks</td></tr>
<tr><td>Practical</td><td>Theoretical</td><td>Practical</td><td>Theoretical</td></tr>
<tr><td>30</td><td>-----</td><td>20</td><td>----</td></tr>
</table>

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Develop and Execute simple Python programs.*
2. *Structure a Python program into functions.*
3. *Using Python lists, tuples to represent compound data*
4. *Develop Python Programs for file processing*

Course Content:

Practical

UNIT I. Introduction to Python, Python, Features of Python, Execution of a Python, Program, Writing Our First Python Program, Data types in Python. Python Interpreter and Interactive Mode; Values and Types: int, float, boolean, string, and list; Variables, Expressions, Statements, Tuple Assignment, Precedence of Operators, Comments; Modules and Functions, Function Definition and use, Flow of Execution, Parameters and Arguments

UNIT II. Operators in Python, Input and Output, Control Statements. Boolean Values and operators, Conditional (if), Alternative (if-else), Chained Conditional (if-elif-else); Iteration: state, while, for, break, continue, pass; Fruitful Functions: Return Values, Parameters, Local and Global Scope, Function Composition, Recursion

UNIT III. Arrays in Python, Strings and Characters. Strings: String Slices, Immutability, String Functions and Methods, String Module; Lists as Arrays. Illustrative Programs: Square Root, gcd, Exponentiation, Sum an Array of Numbers, Linear Search, Binary Search.

UNIT IV. Functions, Lists and Tuples. List Operations, List Slices, List Methods, List Loop, Mutability, Aliasing, Cloning Lists, List Parameters; Tuples: Tuple Assignment, Tuple as Return Value; Dictionaries: Operations and Methods; Advanced List Processing - List Comprehension; Illustrative Programs: Selection Sort, Insertion Sort, Merge sort, Histogram.

UNIT V. Files and Exception: Text Files, Reading and Writing Files, Format Operator; Command Line Arguments, Errors and Exceptions, Handling Exceptions, Modules, Packages; Illustrative Programs: Word Count, Copy File.

The students are required to verify their ability to use core programming basics and program design with functions using Python programming language. The teacher shall programs to strengthen the practical expertise of the students. The following is an indicative list of programs that can be practised.

1. Write a program to demonstrate different number data types in Python.
2. Write a program to perform different Arithmetic Operations on numbers in Python.
3. Write a program to create, concatenate and print a string and accessing sub-string from a given string.
4. Write a python script to print the current date in the following format “Sat Oct 11 02:26:23 IST 2020”
5. Write a program to create, append, and remove lists in python.
6. Write a program to demonstrate working with tuples in python.
7. Write a program to demonstrate working with dictionaries in python.
8. Write a python program to find largest of three numbers.
9. Write a Python program to construct the different pattern, using a nested for loop, Like

```
*
* *
* * *
* *
 *
```

10. Write a Python script that prints prime numbers less than 20.

11. Write a python program to define a module to find Fibonacci Numbers and import the module to another program.
12. Write a python program to define a module and import a specific function in that module to another program.
13. Write a program that inputs a text file. The program should print all of the unique words in the file in alphabetical order.
14. Write a Python class to convert an integer to a roman numeral.
15. Write a Python class to reverse a string word by word.

Internal (CA) Evaluation: Practical Note Book (15 marks), Two experiments (10 marks) – one from Part A and another from Part B, Viva-voce (5 marks)

ESE Evaluation: Two experiments (10 marks) – one from Part A and another from Part B, Viva-voce (10 marks)

References/ Suggested Readings:

1. Alex Martelli, Python in a Nutshell, Oreilly Publication.
2. Allen Downey, Think Python, Green Tea Press.
3. Wesley J. Chun, Core Python Programming, Pearson Education.
4. Mark Lutz, Learning Python, Oreilly Publication.
5. Kenneth A. Lambert, Fundamentals of Python: First Programs, Course Technology Inc.

Semester- IV

Course Name: Software Engineering

Course Code: BSCHCOSC401

Course Type: **Core (Theoretical)**	Course Details: **CC-8**			L-T-P: **5 - 1 - 0**	
Credit: **6**	Full Marks: **50**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		….	**10**	….	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Basic knowledge and understanding of the analysis and design of complex systems.*
2. *Ability to apply software engineering principles and techniques.*
3. *To produce efficient, reliable, robust and cost-effective software solutions.*
4. *Ability to work as an effective member or leader of software engineering teams.*
5. *To manage time, processes and resources effectively by prioritizing competing demands to achieve personal and team goals Identify and analyzes the common threats in each domain.*

Course Content:

Theory

UNIT I. Software Development Approaches: Introduction; Evolving Role of Software; Software Characteristics; Software Applications. Software Design Processes: Introduction; What is Meant by Software Engineering? Definitions of Software Engineering; The Serial or Linear Sequential Development Model; Iterative Development Model; The incremental Development Model

UNIT II. Software Design Principles: Introduction, System Models: Data-flow Models, Semantic Data Models, Object Models, Inheritance Models, Object Aggregation, Service Usage Models, Data Dictionaries; Software Design: The Design Process, Design Methods, Design description, Design Strategies, Design Quality; Architectural Design: System Structuring, The Repository Model, The Client–Server Model, The Abstract Machine Model, Control Models, Modular Decomposition, Domain-Specific Architectures.

UNIT III. Object Oriented Design: Introduction; Object Oriented Design: Objects, Object Classes & Inheritance, Inheritance, Object Identification, An Object -Oriented Design Example, Object Aggregation; Service Usage; Object Interface Design: Design Evolution, Function Oriented Design, Data–Flow Design; Structural Decomposition: Detailed Design.

UNIT IV. An Assessment of Process Life-Cycle Models: Introduction; Overview of the Assessment of Process; The Dimension of Time; The Need for a Business Model in Software Engineering; Classic Invalid Assumptions: First Assumption: Internal or External Drivers,

Second Assumption: Software or Business Processes, Third Assumption: Processes or Projects, Fourth Assumption: Process Centered or Architecture Centered; Implications of the New Business Model; Role of the Problem - Solving Process in this Approach: Data, Problem Definition, Tools and Capabilities; Redefining the Software Engineering Process: Round-Trip Problem-Solving Approach, Activities, Goals, Interdisciplinary Resources, Time.

UNIT V. Software Reliability: Introduction; Software Reliability Metrics; Programming for Reliability: Fault Avoidance, Fault Tolerance, Software Reuse.

UNIT VI. Software Testing Techniques: Introduction; Software Testing Fundamental; Testing Principles; White Box Testing; Control Structure Testing; Black Box Testing; Boundary Value Analysis; Testing GUIs; Testing Documentation and Help Facilities; Software Testing Strategies: Introduction; Organizing for Software Testing; Software Testing Strategy, Unit Testing: Unit Test Considerations, Top-Down Integration, Bottom-Up Integration.

References/ Suggested Readings:

1. R. G. Pressman, Software Engineering, TMH.
2. Rajib Mall, Fundamentals of Software Engineering, PHI Learning Pvt. Ltd.
3. Sommerville, Ian, Software Engineering, Pearson Education
4. Pankaj Jalote, An Integrated Approach to Software Engineering, Narosa Publications.
5. Pfleeger, Shari Lawrence, Software Engineering Theory and Practice, Second edition. Prentice- Hall 2001.

Course Name: Database Management System

Course Code: BSCHCOSC402

<table>
<tr><td colspan="2">Course Type: Core (Theoretical & Practical)</td><td colspan="2">Course Details: CC-9</td><td colspan="2">L-T-P: 4 - 0 - 4</td></tr>
<tr><td rowspan="3">Credit: 6</td><td rowspan="3">Full Marks: 100</td><td colspan="2">CA Marks</td><td colspan="2">ESE Marks</td></tr>
<tr><td>Practical</td><td>Theoretical</td><td>Practical</td><td>Theoretical</td></tr>
<tr><td>30</td><td>10</td><td>20</td><td>40</td></tr>
</table>

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Gain knowledge of database systems and database management systems software*
2. *Ability to model data in applications using conceptual modelling tools such as ER Diagrams and design data base schemas based on the model.*
3. *Formulate, using SQL, solutions to a broad range of query and data update problems.*
4. *Demonstrate an understanding of normalization theory and apply such knowledge to the normalization of a database.*

5. *Be acquainted with the basics of transaction processing and concurrency control.*
6. *Familiarity with database storage structures and access techniques.*
7. *Compare, contrast and analyse the various emerging technologies for database systems.*
8. *Analyse strengths and weaknesses of the applications of database technologies to various subject areas*

Course Content:

Theory

UNIT I. Basic Database Concepts, Terminology, and Architecture; Types of Database Management Systems. Differences between Relational and other Database Models. Data Modelling: Relations, Schemas, Constraints, Queries, and Updates; Conceptual vs. Physical Modeling; Entity Types, attributes, ER Diagrams.

UNIT II. SQL Data Definition: Specifying Tables, Data Types, Constraints; Simple SELECT, INSERT, UPDATE, DELETE Statements; Complex SELECT Queries, including Joins and Nested Queries; Actions and Triggers; Views; Altering Schemas.

UNIT III. Relational Algebra: Definition of Algebra; Relations as Sets; Operations: SELECT, PROJECT, JOIN, etc. Normalization Theory and Functional Dependencies, 2NF, 3NF, BCNF, 4NF, 5NF;

UNIT IV. Indexing: Files, Blocks, and Records, Hashing; RAID; Replication; Single-Level and Multi-Level Indexes; B-Trees and B+-Trees. Basics of Transactions, Concurrency and Recovery.

UNIT V. DATABASE PROGRAMMING: Embedded SQL; Dynamic SQL, Avoiding Injection Attacks; Stored Procedures;

UNIT VI. BIG DATA: Motivations; OLAP vs. OLTP; Batch Processing; Map Reduce and Hadoop; Spark;

Practical

Students are required to practice the concepts learnt in the theory by designing and querying a database for a chosen organization (Like Library, Transport etc). The teacher may devise appropriate weekly lab assignments to help students practice the designing, querying a database in the context of example database. Some indicative list of experiments is given below.

Experiment 1: E-R Model Analyze the organization and identify the entities , attributes and relationships in it. . Identify the primary keys for all the entities. Identify the other keys like candidate keys, partial keys, if any.

Experiment 2: Concept design with E-R Model Relate the entities appropriately. Apply cardinalities for each relationship. Identify strong entities and weak entities (if any).

Experiment 3: Relational Model Represent all the entities (Strong, Weak) in tabular fashion. Represent relation ships in a tabular fashion.

Experiment 4: Normalization Apply the First, Second and Third Normalization levels on the database designed for the organization

Experiment 5: Installation of Mysql/Postgresql/SQL Server/Oracle

Experiment 6: Practicing DDL commands. Creating databases, How to create tables, altering the database, dropping tables and databases. Try truncate, rename commands etc.

Experiment 7: Practicing DML commands on the Database created for the example organization DML commands are used to for managing data within schema objects. Some examples: ● SELECT - retrieve data from the a database ● INSERT - insert data into a table ● UPDATE - updates existing data within a table ● DELETE - deletes all records or few records from a table.

Experiment 8: Querying practice queries (along with sub queries) involving ANY, ALL, IN, Exists, NOT EXISTS, UNION, INTERSECT, Constraints etc.

Experiment 9: Practice queries using Aggregate functions (COUNT, SUM, AVG, and MAX and MIN), GROUP BY, HAVING and Creation and dropping of Views.

Experiment 10: Triggers - Work on Triggers. Creation of, insert trigger, delete trigger, update trigger.

Internal (CA) Evaluation: Practical Note Book (15 marks), Experiment (10 marks) – Experiment 1 to Experiment 5, Viva-voce (5 marks)

ESE Evaluation: Experiment (10 marks) – Experiment 6 to Experiment10, Viva-voce (10 marks)

References/ Suggested Readings:

1. Elmasri's and Navathe's Fundamentals of Database Systems. Addison-Wesley
2. Raghu Ramakrishnan, Johannes Gehrke, Data base Management Systems, McGraw Hill Education
3. Silberschatz, Henry. F. Korth, S. Sudarshan, Data base System Concepts, McGraw Hill Education
4. A. Majumdar and P. Bhattacharyya, Database Management Systems, McGraw Hill Education.

Course Name: Object Oriented Programming

Course Code: BSCHCOSC403

Course Type: **Core (Theoretical & Practical)**	Course Details: **CC-10**			L-T-P: **4 - 0 - 4**	
Credit: **6**	Full Marks: **100**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		30	**10**	**20**	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Learn the concepts of data, abstraction and encapsulation*
2. *Be able to write programs using classes and object.*
3. *Understand conceptually principles of Inheritance and Polymorphism and their use and program level implementation.*
4. *Learn exception and basic event handling mechanisms in a program.*
5. *Learn generic programming with class templates and function templates.*
6. *To learn typical object-oriented constructs of specific object oriented programming language*

Course Content:

Theory

UNIT I. Basics: Introduction to Object Oriented Programming and its Basic Features, Basic Components of C++, Characteristics of Object-Oriented Language, Structure of a C++ Program, Flow Control Statements in C++, Functions - Scope of Variables, Inline Functions, Recursive Functions, Pointers to Functions, C++ Pointers, Arrays, Dynamic Memory Allocation and De-Allocation

UNIT II. Differences Between Object Oriented and Procedure Oriented Programming, Abstraction, Overview of Object-Oriented Programming Principles, Encapsulation, C++ Classes, Objects, User Defined Types, Constructors and Destructors, this Pointer, Friend Functions, Data Abstraction, Operator Overloading, Type Conversion

UNIT III. Class Inheritance, Base and Derived Classes, Virtual Base Class, Virtual Functions, Polymorphism, Static and Dynamic Bindings, Base and Derived Class Virtual Functions, Dynamic Binding through Virtual Functions, Pure Virtual Functions, Abstract Classes, Virtual Destructors

UNIT IV. Stream Classes Hierarchy, Stream I/O, File Streams, Overloading the Extraction and Insertion Operators, Error Handling during File Operations, Formatted I/O.

UNIT V. Exception Handling- Benefits of Exception Handling, Throwing an Exception, the Try Block, Catching an Exception, Exception Objects, Exception Specifications, Rethrowing an Exception, Uncaught Exceptions.

UNIT VI. Templates - Class Templates and Function Templates, simple generic classes and generic function, simple example programs. Introduction to Standard Template Library (STL), Components of STL, STL-List, Vector, Array.

Practical

Students are required to understand the object-oriented concepts using C++. They are required to practice the concepts learnt in the theory. Some of the programs to be implemented are listed as follows:

1. Number of vowels and number of characters in a string.
2. Write a function called zeros maller() that is passed with two introduce arguments by reference and set the smaller of the number to zero. Write a main() program to access this function.
3. Demonstration of Class, Constructors, destructors, input and output functions, Objects
4. Demonstration of array of object.
5. Demonstration of friend functions.
6. Demonstration of operator overloading.
7. Demonstration of inheritance.
8. Using this pointer to return a value (return by reference).
9. Demonstration of virtual function.
10. Demonstration of static function.
11. Accessing a particular record in a student's file.
12. Demonstration of exception handling.
13. Demonstration of class template and function template
14. Demonstration of Standard Template Library (STL).

Internal (CA) Evaluation: Practical Note Book (15 marks), Two experiments (10 marks), Viva-voce (5 marks).

ESE Evaluation: Two experiments (10 marks), Viva-voce (10 marks)

References/ Suggested Readings:

1. E.Balagurusamy , Object Oriented Programming through C++,TMH.
2. Lafore Robert, Object Oriented Programming in Turbo C++, Galgotia Publications.
3. Herbert Schildt, C++: The Complete Reference, 4th Edition
4. B. Stroutstrup, The C++ Programming Language, 3rd Edition, Pearson Education.
5. Ashok N Kamthane, Programming in C++, Pearson 2nd Edition.

Course Name: Object Oriented Programming in C++

Course Code: BSCHCOSGE401

Course Type: **GE (Theoretical & Practical)**	Course Details: **GEC-4**			L-T-P: **4 - 0 - 4**	
Credit: **6**	Full Marks: **100**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		30	**10**	**20**	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Learn the concepts of data, abstraction and encapsulation*
2. *Be able to write programs using classes and objects.*
3. *Understand conceptually principles of Inheritance and Polymorphism and their use and program level implementation.*
4. *Learn exception and basic event handling mechanisms in a program*
5. *Learn generic programming with class templates and function templates.*
6. *To learn typical object-oriented constructs of specific object oriented programming language.*

Course Content:

Theory

UNIT I. Basics: Introduction to Object Oriented Programming and its Basic Features, Basic Components of C++, Characteristics of Object-Oriented Language, Structure of a C++ Program, Flow Control Statements in C++, Functions - Scope of Variables, Inline Functions, Recursive Functions, Pointers to Functions, C++ Pointers, Arrays, Dynamic Memory Allocation and De-Allocation

UNIT II. Differences Between Object Oriented and Procedure Oriented Programming, Abstraction, Overview of Object-Oriented Programming Principles, Encapsulation, C++ Classes, Objects, User Defined Types, Constructors and Destructors, this Pointer, Friend Functions, Data Abstraction, Operator Overloading, Type Conversion

UNIT III. Class Inheritance, Base and Derived Classes, Virtual Base Class, Virtual Functions, Polymorphism, Static and Dynamic Bindings, Base and Derived Class Virtual Functions, Dynamic Binding through Virtual Functions, Pure Virtual Functions, Abstract Classes, Virtual Destructors

UNIT IV. Stream Classes Hierarchy, Stream I/O, File Streams, Overloading the Extraction and Insertion Operators, Error Handling during File Operations, Formatted I/O.

UNIT V. Exception Handling- Benefits of Exception Handling, Throwing an Exception, the Try Block, Catching an Exception, Exception Objects, Exception Specifications.

UNIT VI. Templates - Class Templates and Function Templates, simple generic classes and generic function, simple example programs. Introduction to Standard Template Library (STL).

Practical

Students are required to understand the object-oriented concepts using C++. They are required to practice the concepts learnt in the theory. Some of the programs to be implemented are listed as follows:

1. Number of vowels and number of characters in a string.
2. Write a function called zeros maller () that is passed with two introduce arguments by reference and set the smaller of the number to zero. Write a main() program to access this function.
3. Demonstration of Class, Constructors, destructors, input and output functions, Objects
4. Demonstration of array of object.
5. Demonstration of friend functions.
6. Demonstration of operator overloading.
7. Demonstration of inheritance.
8. Using this pointer to return a value (return by reference).
9. Demonstration of virtual function.
10. Demonstration of static function.
11. Accessing a particular record in a student's file.
12. Demonstration of exception handling.
13. Demonstration of class template and function template

Internal (CA) Evaluation: Practical Note Book (15 marks), Two experiments (10 marks), Viva-voce (5 marks)

ESE Evaluation: Two experiments (10 marks), Viva-voce (10 marks)

References/ Suggested Readings:

1. E.Balagurusamy , Object Oriented Programming through C++,TMH.
2. Lafore Robert, Object Oriented Programming in Turbo C++, Galgotia Publications.
3. Herbert Schildt, C++: The Complete Reference, 4th Edition.
4. B. Stroutstrup, The C++ Programming Language, 3rd Edition, Pearson Education.

Course Name: Introduction to Database Management System

Course Code: BSCHCOSGE402

Course Type: **GE (Theoretical & Practical)**	Course Details: **GEC-4**			L-T-P: **4 - 0 - 4**	
Credit: **6**	Full Marks: **100**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		30	**10**	**20**	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Gain knowledge of database systems and database management systems software*
2. *Ability to model data in applications using conceptual modelling tools such as ER Diagrams and design data base schemas based on the model.*
3. *Formulate, using SQL, solutions to a broad range of query and data update problems.*
4. *Demonstrate an understanding of normalization theory and apply such knowledge to the normalization of a database.*
5. *Be acquainted with the basics of transaction processing and concurrency control.*
6. *Familiarity with database storage structures and access techniques.*
7. *Compare, contrast and analyse the various emerging technologies for database systems.*
8. *Analyse strengths and weaknesses of the applications of database technologies to various subject areas*

Course Content:

Theory

UNIT I. Basic Database Concepts, Terminology, and Architecture; Types of Database Management Systems. Differences between Relational and other Database Models. Data Modelling: Relations, Schemas, Constraints, Queries, and Updates; Conceptual vs. Physical Modeling; Entity Types, attributes, ER Diagrams.

UNIT II. SQL Data Definition: Specifying Tables, Data Types, Constraints; Simple SELECT, INSERT, UPDATE, DELETE Statements; Complex SELECT Queries, including Joins and Nested Queries; Actions and Triggers; Views; Altering Schemas.

UNIT III. Relational Algebra: Definition of Algebra; Relations as Sets; Operations: SELECT, PROJECT, JOIN, etc. Normalization Theory and Functional Dependencies, 2NF, 3NF, BCNF, 4NF, 5NF;

UNIT IV. Indexing: Files, Blocks, and Records, Hashing; RAID; Replication; Single-Level and Multi-Level Indexes; B-Trees and B+-Trees. Basics of Transactions, Concurrency and Recovery.

UNIT V. Introduction to BIG DATA: Motivations; Applications.

Practical

Students are required to practice the concepts learnt in the theory by designing and querying a database for a chosen organization (Like Library, Transport etc). The teacher may devise appropriate weekly lab assignments to help students practice the designing , querying a database in the context of example database. Some indicative list of experiments is given below.

Experiment 1: E-R Model Analyze the organization and identify the entities , attributes and relationships in it. . Identify the primary keys for all the entities. Identify the other keys like candidate keys, partial keys, if any.

Experiment 2: Concept design with E-R Model Relate the entities appropriately. Apply cardinalities for each relationship. Identify strong entities and weak entities (if any).

Experiment 3: Relational Model Represent all the entities (Strong, Weak) in tabular fashion. Represent relation ships in a tabular fashion.

Experiment 4: Normalization Apply the First, Second and Third Normalization levels on the database designed for the organization

Experiment 5: Installation of Mysql/Postgresql/SQL Server/Oracle

Experiment 6: Practicing DDL commands. Creating databases, How to create tables, altering the database, dropping tables and databases. Try truncate, rename commands etc.

Experiment 7: Practicing DML commands on the Database created for the example organization DML commands are used to for managing data within schema objects. Some examples: ● SELECT - retrieve data from the a database ● INSERT - insert data into a table ● UPDATE - updates existing data within a table ● DELETE - deletes all records or few records from a table.

Experiment 8: Querying practice queries (along with sub queries) involving ANY, ALL, IN, Exists, NOT EXISTS, UNION, INTERSECT, Constraints etc.

Experiment 9: Practice queries using Aggregate functions (COUNT, SUM, AVG, and MAX and MIN), GROUP BY, HAVING and Creation and dropping of Views.

Experiment 10: Triggers - Work on Triggers. Creation of, insert trigger, delete trigger, update trigger.

Internal (CA) Evaluation: Practical Note Book (15 marks), Experiment (10 marks) – Experiment 1 to Experiment 5, Viva-voce (5 marks)

ESE Evaluation: Experiment (10 marks) – Experiment 6 to Experiment10, Viva-voce (10 marks)

References/ Suggested Readings:

1. Elmasri's and Navathe's Fundamentals of Database Systems. Addison-Wesley.
2. Raghu Ramakrishnan, Johannes Gehrke, Data base Management Systems, McGraw Hill Education.
3. Silberschatz, Henry. F. Korth, S. Sudarshan, Data base System Concepts, McGraw Hill Education.
4. A. Majumdar and P. Bhattacharyya, Database Management Systems, McGraw Hill Education.

Course Name: Mobile Application Development

Course Code: BSCHCOSSE401

Course Type: **SE (Practical)**	Course Details: **SEC-2**			L-T-P: **0 - 0 - 8**	
Credit: **4**	Full Marks: **50**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		30	**-----**	**20**	**----**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *To understand Android platform and its architecture.*
2. *To learn about mobile devices types and different modern mobile operating systems.*
3. *To learn activity creation and Android User Interface designing.*
4. *To learn basics of Intent, Broadcast and Internet services.*
5. *To learn about different wireless mobile data transmission standards.*
6. *To understand and learn how to integrate basic phone features, multimedia, camera and Location based services in Android Application.*
7. *To learn about different systems for mobile application development, deployment and distribution in Mobile market place (Android, IoS).*
8. *To understand and carry out functional test strategies for mobile applications.*

Course Content:

Practical

Students are directed to do a minor project based on the contents of the course below (UNIT I to UNIT VI) for internal and ESE evaluation.

UNIT I. (Introduction) What is Android, Android Versions and its Feature Set, Various Android Devices on the Market, Android Market Application Store, Android Development Environment System Requirements, Android SDK, Installing Java, and ADT bundle - Eclipse Integrated Development Environment (IDE), Creating Android Virtual Devices (AVDs).

UNIT II. (Android Architecture Overview and Application) Android Software Stack, The Linux Kernel, Android Runtime - Dalvik Virtual Machine, Android Runtime – Core Libraries, Dalvik VM Specific Libraries, Java Interoperability Libraries, Android Libraries, Application Framework, Creating a New Android Project ,Defining the Project Name and SDK Settings, Project Configuration Settings, Configuring the Launcher Icon, Creating an Activity, Running the Application in the AVD, Stopping a Running Application, Modifying the Example Application, Reviewing the Layout and Resource Files.

UNIT III. (Android Software Development Platform and Framework) Understanding Java SE and the Dalvik Virtual Machine, The Directory Structure of an Android Project, Common Default Resources Folders, The Values Folder, Leveraging Android XML, Screen Sizes , Launching Mobile Application: The AndroidManifest.xml File, Android Application Components, Android Activities: Defining the UI, Android Service s: Processing in the Background, Broadcast Receivers: Announcements and Notifications Content Providers: Data Management, Android Intent Objects: Messaging for Components, Android Manifest XML: Declaring Your Components.

UNIT IV. (Understanding Android User Interfaces, Views and Layouts) Designing for Different Android Devices, Views and View Groups, Android Layout Managers, The View Hierarchy, Designing an Android User Interface using the Graphical Layout Tool Displaying Text with TextView, Retrieving Data from Users, Using Buttons, Check Boxes and Radio Groups, Getting Dates and Times from Users, Using Indicators to Display Data to Users, Adjusting Progress with Seek Bar, Working with Menus using views, Gallery, Image Switcher, Grid View, and Image View views to display images, Creating Animation.

UNIT V. (Databases, Intents, Location-based Services) Saving and Loading Files, SQLite Databases, Android Database Design, Exposing Access to a Data Source through a Content Provider, Content Provider Registration, Native Content Providers Intents and Intent Filters: Intent Overview, Implicit Intents, Creating the Implicit Intent Example Project, Explicit Intents, Creating the Explicit Intent Example Application, Intents with Activities, Intents with Broadcast Receivers.

UNIT VI. Sending SMS Messages Programmatically, Getting Feedback after Sending the Message Sending SMS Messages Using Intent Receiving, sending email, Introduction to location-based service, configuring the Android Emulator for Location -Based Services, Geocoding and Map-Based Activities Multimedia: Audio, Video, Camera: Playing Audio and Video, Recording Audio and Video, Using the Camera to Take and Process Pictures.

Internal (CA) Evaluation: Minor Project Report (15 marks), Demonstration of the minor project (10 marks), Viva-voce (5 marks)

ESE Evaluation: Presentation of the minor project (10 marks), Viva-voce (10 marks).

References/ Suggested Readings:

1. Harwani, Android Programming Unleashed, SAMS.
2. Richard Rodger, Beginning Mobile Application Development in the Cloud, Wrox.

Course Name: Web Programming

Course Code: BSCHCOSSE402

Course Type: **SE (Practical)**	Course Details: **SEC-2**			L-T-P: **0 - 0 - 8**	
Credit: **4**	Full Marks: **50**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		30	**-----**	**20**	**----**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *To understand basics of the Internet and World Wide Web*
2. *To acquire knowledge and skills for creation of web site considering both client and server-side programming*
3. *To learn basic skill to develop responsive web applications*
4. *To understand different web extensions and web services standards*
5. *To understand basic concepts of Search Engine Basics.*
6. *To learn Web Service Essentials.*
7. *To learn Rich Internet Application Technologies.*
8. *To understand and get acquainted with Web Analytics 2.0*

Course Content:

Practical

Students are directed to do a minor project based on the contents of the course below (UNIT I to UNIT VI) for internal and ESE evaluation.

UNIT I. (Introduction to World Wide Web) -Internet Standards, Introduction to WWW and WWW Architecture, Internet Protocols, Overview of HTTP, HTTP request – response, Generations of dynamic web pages.

UNIT II. (User Interface Design) Introduction to HTML and HTML5, TML Tags, Formatting and Fonts, Commenting Code, Anchors, Backgrounds, Images, Hyperlinks, Lists,

Tables, Frames, HTML Forms. The need for CSS, Introduction to CSS, Basic syntax and structure, Inline Styles, Embedding Style Sheets, Linking External Style, Backgrounds, Manipulating Text, Margins and Padding, Positioning using CSS.

UNIT III. (Java Programming) Java Script, Introduction, Core features, Data types and Variables, Operators, Expressions, Functions, Objects, Array, Date and Math related Objects. JAVA Networking classes, TCP/IP Protocol Suite, File Transfer Protocol (FTP), Java Environment, Setup for Web Applications, JavaBean, Application Builder Tool, Bean Developer Kit (BDK), The Java Beans API, Introduction to EJB

UNIT IV. (Database) Database basics, SQL, MySQL, PostgreSQL, JDBC API, Driver Types, Two-tier and Three-tier Models, Connection Overview, Transactions, Driver Manager Overview, Statement Overview, Result Set Overview, Types of Result Sets, Concurrency Types, Prepared Statement Overview

UNIT V. (Java Applet and JSP) Java Web Programs and Applets, Web Application, Servlet, Servlet Life Cycle, Servlet Programming, Introduction to JSP, Life Cycle of a JSP Page, Translation and Compilation, Creating Static Content, Response and Page Encoding, Creating Dynamic Content, Using Objects within JSP Pages, JSP Programming

UNIT VI. (Dot Net Framework) Introduction to Dot Net, Dot Net framework and its architecture, CLR, Assembly, Components of Assembly, DLL hell and Assembly Versioning, Overview to C#, Introduction to ASP.net, Asp.net Programming.

Internal (CA) Evaluation: Minor Project Report (15 marks), Demonstration of the minor project (10 marks), Viva-voce (5 marks).

ESE Evaluation: Presentation of the minor project (10 marks), Viva-voce (10 marks).

References/ Suggested Readings:

1. James Keogh, J2EE: The complete Reference.
2. John Brock, Arun Gupta, Geertjan Wielenga, Java EE and HTML5 Enterprise Application Development (Oracle Press)
3. James Holmes, Struts: The Complete Reference, 2nd Edition
4. Stephen Walther, Kevin Scott Hoffman, Nate Dudek, ASP.NET Unleashed
5. John Sharp, Microsoft Visual C# 2013 Step by Step.
6. A. Majumdar and P. Bhattacharyya, Database Management Systems, McGraw Hill Education.

Semester- V

Course Name: Internet Technologies

Course Code: BSCHCOSC501

Course Type: **Core (Theoretical & Practical)**	Course Details: **CC-11**			L-T-P: **4 - 0 - 4**	
Credit: **6**	Full Marks: **100**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		30	**10**	**20**	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *To understand the terms related to the Internet and how the Internet is changing the world.*
2. *To understand how computers are connected to the Internet and demonstrate the ability to use the World Wide Web.*
3. *Demonstrate an understanding of and the ability to use electronic mail and other internet based services*
4. *Understand the design principles of Web pages and how they are created*
5. *To develop an ability to create basic Web pages with HTML, CSS, JavaScript etc.*

Course Content:

Theory

UNIT I. Introduction: Overview, Network of Networks, Intranet, Extranet and Internet. World Wide Web, Domain and Sub domain, Address Resolution, DNS, Telnet, FTP, HTTP. Review of TCP/IP: Features, Segment, Three-Way Handshaking, Flow Control, Error Control, Congestion control.

UNIT II. IP Datagram, IPv4 and IPv6. IP Subnetting and addressing: Classful and Classless Addressing, Subnetting. NAT, IP masquerading, IP tables. Internet Routing Protocol: Routing -Intra and Inter Domain Routing, Unicast and Multicast Routing, Broadcast. Electronic Mail: POP3, SMTP.

UNIT III. HTML: Introduction, Editors, Elements, Attributes, Heading, Paragraph. Formatting, Link, Head, Table, List, Block, Layout, CSS. Form, Iframe, Colors, Colorname, Colorvalue. Image Maps: map, area, attributes of image area. Extensible Markup Language (XML): Introduction, Tree, Syntax, Elements, Attributes, Validation, Viewing. XHTML in brief. CGI Scripts: Introduction, Environment Variable, GET and POST Methods.

UNIT IV. PERL: Introduction, Variable, Condition, Loop, Array, Implementing data structure, Hash, String, Regular Expression, File handling, I/O handling. JavaScript: Basics, Statements, comments, variable, comparison, condition, switch, loop, break. Object - string,

array, Boolean, reg-ex. Function, Errors, Validation. Cookies: Definition of cookies, Create and Store a cookie with example. Java Applets: Container Class, Components, Applet Life Cycle, Update method; Parameter passing applet, Applications.

UNIT V. Client-Server programming in Java: Java Socket, Java RMI. Threats: Malicious code-viruses, Trojan horses, worms; eavesdropping, spoofing, modification, denial of service attacks. Network security techniques: Password and Authentication; VPN, IP Security, security in electronic transaction, Secure Socket Layer (SSL), Secure Shell (SSH). Firewall: Introduction, Packet filtering, Stateful, Application layer, Proxy.

UNIT VI. Internet Telephony: Introduction, VoIP. Multimedia Applications: Multimedia over IP: RSVP, RTP, RTCP and RTSP. Streaming media, Codec and Plugins, IPTV. Search Engine and Web Crawler: Definition, Meta data, Web Crawler, Indexing, Page rank, overview of SEO.

Practical

Students are directed to do a minor project based on the contents of the course below (UNIT I to UNIT V) for internal and ESE evaluation.

UNIT I. HTML: Introduction to HTML and HTML5, HTML Tags, Formatting and Fonts, Commenting Code, Anchors, Backgrounds, Images, Hyperlinks, Lists, Tables, Frames, HTML Forms.

UNIT II. CSS: The need for CSS, Introduction to CSS, Basic syntax and structure, Inline Styles, Embedding Style Sheets, Linking External Style, Backgrounds, Manipulating Text, Margins and Padding, Positioning using CSS.

UNIT III. JavaScript: Syntax, Variables, Values, Data Types, Data Types, Expressions and Operators, Control structures, Error handling, Throwing errors, Numbers, Strings, Arrays.

UNIT IV. PHP**:** Introduction to PHP**,** Server side scripting, Role of web server software, PHP comments, variables, echo and print, PHP operators, data types , Branching statements, Loops, Arrays, PHP functions, PHP form, Passing information between pages, $_GET, $_POST, $_REQUEST., String functions, include and require, session and cookie management, Error handling in PHP, Object Oriented Programming using PHP.

UNIT V. PHP with MYSQL: Introduction to MySQL, datatypes, SQL commands-CREATE, UPDATE, INSERT, DELETE, SELECT, PHP functions for MySQL connectivity and operation- mysql_connect, mysql_select_db, mysql_query, Updation and deletion of data using PHP, Displaying data from MySQL in webpage.

Internal (CA) Evaluation: Minor Project Report (15 marks), Demonstration of the minor project (10 marks), Viva-voce (5 marks).

ESE Evaluation: Presentation of the minor project (10 marks), Viva-voce (10 marks).

References/ Suggested Readings:

1. N.P. Gopalan and J. Akilandeswari, Web Technology: A Developer's Perspective, PHI.
2. Rahul Banerjee, Internetworking Technologies, An Engineering Perspective, PHI.
3. Susan Anderson-Freed, Weaving a Website: Programming in Html, Java Script, Perl and Java, Prentice Hall.
4. P.J. Deitel & H.M. Deitel, Internet and World Wide Web How to program, Pearson.
5. Ivan Bayross ,HTML, DHTML, JavaScript, Pearl & CGI, Fourth Revised Edition, BPB Publication.
6. Rasmus Lerdorf and Kevin Tatore, Programming PHP, Shroff Publishers & Distributors Pvt. Ltd.
7. Dave W Mercer, Allan Kent, Steven D Nowicki, David Mercer, Dan Squier, Wankyu Choi, Beginning PHP, Wiley Publishing, In.
8. Robin Nixon, Learning Php, MySQL & JavaScript: A Step-By-Step Guide To Creating Dynamic Websites, O'REILLY.

Course Name: Artificial Intelligence

Course Code: BSCHCOSC502

<table>
<tr><td rowspan="2">Course Type: Core (Theoretical & Practical)</td><td colspan="2" rowspan="2">Course Details: CC-12</td><td colspan="2" rowspan="2">L-T-P: 4 - 0 - 4</td></tr>
<tr></tr>
<tr><td rowspan="3">Credit: 6</td><td rowspan="3">Full Marks: 100</td><td colspan="2">CA Marks</td><td colspan="2">ESE Marks</td></tr>
<tr><td>Practical</td><td>Theoretical</td><td>Practical</td><td>Theoretical</td></tr>
<tr><td>30</td><td>10</td><td>20</td><td>40</td></tr>
</table>

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Explain what constitutes "Artificial" Intelligence and how to identify systems with Artificial Intelligence.*
2. *Identify problems that are amenable to solution by AI methods, and which AI methods may be suited to solving a given problem.*
3. *Formalize a given problem in the language/framework of different AI methods (e.g., as a search problem, as a constraint satisfaction problem, as a planning problem, etc).*
4. *Implement basic AI algorithms (e.g., standard search or constraint propagation algorithms).*
5. *Design and perform an empirical evaluation of different algorithms on a problem formalisation, and state the conclusions that the evaluation supports.*
6. *Explain the limitations of current Artificial Intelligence techniques.*

Course Content:

Theory

UNIT I. Introduction to Artificial Intelligence: Definition of AI; Turing Test; Brief History of AI. Problem Solving and Search: Problem Formulation; Search Space; States vs. Nodes; Tree Search: Breadth-First, Uniform Cost, Depth-First, Depth-Limited, Iterative Deepening; Graph Search.

UNIT II. Informed Search: Greedy Search; A* Search; Heuristic Function; Admissibility and Consistency; Deriving Heuristics via Problem Relaxation. Local Search: Hill-Climbing; Simulated Annealing; Genetic Algorithms; Local Search in Continuous Spaces.

UNIT III. Playing Games: Game Tree; Utility Function; Optimal Strategies; Minimax Algorithm; Alpha-Beta Pruning; Games with an Element of Chance. Beyond Classical Search: Searching with Nondeterministic Actions; Searching with Partial Observations; Online Search Agents; Dealing with Unknown Environments.

UNIT IV. Knowledge Representation and Reasoning: Ontologies, Foundations of Knowledge Representation and Reasoning, Representing and Reasoning about Objects, Relations, Events, Actions, Time, and Space; Predicate Logic, Situation Calculus, Description Logics, Reasoning with Defaults, Reasoning about Knowledge, Sample Applications.

UNIT V. Representing and Reasoning with Uncertain Knowledge: Probability, Connection to Logic, Independence, Bayes Rule, Bayesian Networks, Probabilistic Inference, and Sample Applications.

UNIT VI. Planning: The STRIPS Language; Forward Planning; Backward Planning; Planning Heuristics; Partial-Order Planning; Planning using Propositional Logic; Planning vs. Scheduling.

UNIT VII. Constraint Satisfaction Problems (CSPs): Basic Definitions; Finite vs. Infinite vs. Continuous Domains; Constraint Graphs; Relationship With Propositional Satisfiability, Conjunctive Queries, Linear Integer Programming, and Diophantine Equations; NP49 Completeness of CSP; Extension to Quantified Constraint Satisfaction (QCSP). Constraint Satisfaction as a Search Problem; Backtracking Search; Variable and Value Ordering Heuristic; Degree Heuristic; Least-Constraining Value Heuristic; Forward Checking; Constraint Propagation; Dependency-Directed Backtracking;

Practical

UNIT I. LISP:

1. **Introduction**
   The need for symbolic computation. Why LISP is a good language for symbolic computation.
2. **LISP Introduction**
   Atoms, lists, S-expressions, functions, lambdas, predicates, conditionals, recursion, iteration, printing, reading, properties, a-lists.

3. **Search: The General Problem Solver**
   Means-ends analysis, defining operators, blocks-world planning, Sussman Anomaly, interacting goals.
4. **Pattern Matching: ELIZA**
   Pattern matching, rule-based translation, a simplistic natural-language dialog system.
5. **Search Tools**
   A general search program. Heuristic search, best-first search, beam-search, hill-climbing.
6. **Advanced LISP**
   Macros, lexical scoping, lexical closures, special variables, dynamic scoping, multiple values. Consing, destructive functions, and garbage collection. Interning symbols. Caching. Delaying computation. Avoiding unnecessary consing.

UNIT II. Logic Programming in Prolog:

1. **Introduction to Prolog.** The structure of a Prolog program and how to use the Prolog interpreter. Unification. Some simple programs.
2. **Arithmetic and lists**. Prolog's support for evaluating arithmetic expressions and lists.
3. **Backtracking, cut, and negation.** The cut operator for controlling backtracking. Negation as failure and its uses.
4. **Search and cut.** Prolog's search method for solving problems. Graph searching exploiting Prolog's built-in search mechanisms.
5. **Difference structures**. Difference lists: introduction and application to example programs.

Internal (CA) Evaluation: Practical Note Book (15 marks), Two Experiments (10 marks) – One from UNIT I and another from UNIT II, Viva-voce (5 marks).

ESE Evaluation: Two Experiments (10 marks) – One from UNIT I and another from UNIT II, Viva-voce (10 marks).

References/ Suggested Readings:

1. Elaine Rich, Kevin Knight, Shivashankar B Nair, Artificial Intelligence, McGraw Hill Edition.
2. Russell Stuart Jonathan and Norvig Peter, Artificial Intelligence: A Modern Approach, Prentice Hall.
3. Dan W. Patterson, Introduction to Artificial Intelligence and Expert Systems, PHI.
4. Paul Graham, ANSI Common Lisp, Prentice Hall.
5. I. Bratko, PROLOG programming for artificial intelligence. Addison-Wesley.
6. L. Sterling, and E. Shapiro, The art of Prolog. MIT Press.

Course Name: Image Processing

Course Code: BSCHCOSDSE501

Course Type: **DSE (Theoretical)**	Course Details: **DSEC-1 or DSEC-2**			L-T-P: **5 - 1 - 0**	
Credit: **6**	Full Marks: **50**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		….	**10**	….	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *To familiarize the students with the image fundamentals and mathematical transforms necessary for image processing.*
2. *To make the students understand the image enhancement techniques*
3. *To make the students understand the image restoration and reconstruction procedures.*
4. *To familiarize the students with the image segmentation procedures.*

Course Content:

Theory

UNIT I. Digital Image Fundamentals: Elements of Visual Perception, Light, Brightness Adaption and Discrimination, Image Sensing and Acquisition, Image Sampling and Quantization, Pixels, Some Basic Relationships between Pixels, Coordinate Conventions, Imaging Geometry, Perspective Projection, Linear and Nonlinear Operations.

UNIT II. Image Enhancement in the Spatial Domain: Intensity transformations, Contrast Stretching, Histogram Equalization, Correlation and Convolution, Basics of Spatial Filtering, Smoothing Filters, Sharpening Filters, Gradient and Laplacian.

UNIT III. Filtering in the Frequency domain: Fourier Transforms [one-dimensional Discrete Fourier Transform (DFT), two-dimensional DFT] and properties, FFT (Decimation in Frequency and Decimation in Time Techniques), Convolution, Correlation, 2-D sampling, Discrete Cosine Transform, Discrete Sine Transform, Discrete Wavelet Transform of Images (Haar and Hadamard), Hotelling/KL Transform, Frequency domain filtering [low pass filters, high pass filters and band pass filters].

UNIT IV. Image Restoration and Reconstruction: Basic Framework, Interactive Restoration, Image deformation and geometric transformations, image morphing, Restoration techniques, Noise characterization, Noise restoration filters, Adaptive filters, Linear, Position invariant degradations, Estimation of Degradation functions, Restoration from projections.

UNIT V. Color Image Processing, Color Fundamentals, Color Models, Pseudo color Image Processing, Basics of Full-Color Image Processing, Color Transformations, Smoothing and

Sharpening, Color Segmentation. Morphological Image Processing, Dilation and Erosion, Opening and Closing., Extensions to Gray -Scale Images.

UNIT VI. Image Segmentation: Detection of Discontinuities, Edge Linking and Boundary Detection, Thresholding, Region-Based Segmentation, Segmentation by Morphological Watersheds.

References/ Suggested Readings:

1. Rafael C. Gonzalez and Richard E. Woods, Digital Image Processing, 4th Edition, Prentice Hall.
2. Anil K. Jain, Fundamentals of Digital Image Processing, Prentice Hall.
3. Stan Birchfield, Image Processing and Analysis, Cengage Learning.
4. B. Chanda and D. Dutta Majumder, Digital Image Processing and Analysis, Prentice Hall of India

Course Name: Data Analytics

Course Code: BSCHCOSDSE502

Course Type: **DSE (Theoretical)**	Course Details: **DSEC-1 or DSEC-2**			L-T-P: **5 - 1 - 0**	
Credit: **6**	Full Marks: **50**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		….	**10**	….	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *This course prepares students to gather, describe, and analyze data, and use advanced statistical tools to support decision making.*
2. *To gather sufficient relevant data, conduct data analytics using scientific methods, and understand appropriate connections between quantitative analysis and realworld problems.*
3. *Understand the exact scopes and possible limitations of each method to provide constructive guidance in decision making.*
4. *To Use advanced techniques to conduct thorough and insightful analysis, and interpret the results correctly with detailed and useful information.*
5. *To make better decisions by using advanced techniques in data analytics.*

Course Content:

Theory

UNIT I. Data Definitions and Analysis Techniques: Elements, Variables, and Data Categorization, Levels of Measurement, Data Management and Indexing.

UNIT II. Descriptive Statistics: Measures of Central Tendency, Measures of Location of Dispersions, Error Estimation and Presentation (Standard Deviation, Variance), Introduction to Probability.

UNIT III. Basic Analysis Techniques: Statistical Hypothesis Generation and Testing, Chi-Square Test, T-Test, Analysis of Variance, Correlation Analysis, Maximum Likelihood Test

UNIT IV. Data Analysis Techniques-I: Regression Analysis, Classification Techniques, Clustering Techniques (K-Means, K-Nearest Neighborhood).

UNIT V. Data Analysis Techniques-II: Association Rules Analysis, Decision Tree.

UNIT VI. Introduction to R Programming: Introduction to R Software Tool, Statistical Computations using R (Mean, Standard Deviation, Variance, Regression, Correlation etc.)

UNIT VII. Practice and Analysis with R, Sensitivity Analysis.

References/ Suggested Readings:

1. Ronald E Walppole, Raymond H Myres, Sharon L. Myres and Leying Ye,,Probability and statistics for Engineers and Scientists, Prentice Hall Inc
2. Travor Hastie Robert Tibshirani Jerome Friedman, The Elements of Statistical Learning, Data Mining, Inference, and Prediction, Springer.
3. John M. Chambers, Software for Data Analysis: Programming with R (Statistics and Computing), Springer

Course Name: Computer Ethics

Course Code: BSCHCOSDSE503

<table>
<tr><td rowspan="2">Course Type: DSE (Theoretical)</td><td colspan="2" rowspan="2">Course Details: DSEC-1 or DSEC-2</td><td colspan="2">L-T-P: 5 - 1 - 0</td></tr>
<tr><td colspan="2"></td></tr>
<tr><td rowspan="3">Credit: 6</td><td rowspan="3">Full Marks: 50</td><td colspan="2">CA Marks</td><td colspan="2">ESE Marks</td></tr>
<tr><td>Practical</td><td>Theoretical</td><td>Practical</td><td>Theoretical</td></tr>
<tr><td>….</td><td>10</td><td>….</td><td>40</td></tr>
</table>

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *The student will be able to describe and distinguish between the various ethical theories which can be used to form the basis of solutions to moral dilemmas in computing.*
2. *Identify traditional and current Issues related to Computers, Information Systems, Ethics, Society and Human Values;*

3. *The student will be able to identify and define the components of a structured plan for solving ethical problems and, in the process, will be able to understand the basis for her/his own ethical system.*
4. *Given several examples of professional codes of ethics related to computing, the student will be able to compare and contrast these examples, discussing their commonalties, differences, and implications.*
5. *Develop skills of critical analysis and applying ethical principles to situations and dialectical thinking.*

Course Content:

Theory

UNIT I. The Need for Computer Ethics Training and Historical Milestones.

UNIT II. Defining the Field of Computer Ethics, Computer ethics codes, Sample Topics in Computer Ethics

i. Computer crime and computer security
ii. Software theft and intellectual property rights
iii. Computer hacking and the creation of viruses
iv. Computer and information system failure
v. Invasion of privacy. Privacy in the Workplace and on the Internet
vi. Social implications of artificial intelligence and expert systems
vii. The information technology salesman issues.

UNIT III. Transparency and Virtual Ethics, Free Speech, Democracy, Information Access.

UNIT IV. Developing the Ethical Analysis Skills and Professional Values, Privacy, Accountability, Government Surveillance.

UNIT V. Boundaries of Trust, Trust Management, Wikipedia, Virtual Trust, Plagiarism in Online Environment, Intellectual Property, Net neutrality.

References/ Suggested Readings:

1. J. Deborah, H. Nissenbaun, Computing, Ethics & Social Values, Englewwod Cliffs, New Jersey, Prentice Hall.
2. R. Spinello, H. T. Tavani, Readings in Cyberethics, Sudbury, MA, Jones and Bartlett Publishers.
3. T. W. Bynum, S. Rogerson, S, Computer Ethics and Professional Responsibility, Blackwell.

Course Name: System Security

Course Code: BSCHCOSDSE504

<table>
<tr><td>Course Type: DSE (Theoretical)</td><td colspan="2">Course Details: DSEC-1 or DSEC-2</td><td colspan="2">L-T-P: 5 - 1 - 0</td></tr>
<tr><td rowspan="3">Credit: 6</td><td rowspan="3">Full Marks: 50</td><td colspan="2">CA Marks</td><td colspan="2">ESE Marks</td></tr>
<tr><td>Practical</td><td>Theoretical</td><td>Practical</td><td>Theoretical</td></tr>
<tr><td>….</td><td>10</td><td>….</td><td>40</td></tr>
</table>

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Develop an understanding of information assurance as practiced in computer operating systems, distributed systems, networks and representative applications.*
2. *Gain familiarity with prevalent network and distributed system attacks, defenses against them, and forensics to investigate the aftermath.*
3. *Develop a basic understanding of cryptography, how it has evolved, and some key encryption techniques used today.*
4. *Develop an understanding of security policies (such as authentication, integrity and confidentiality), as well as protocols to implement such policies in the form of message exchanges.*

Course Content:

Theory

UNIT I. Cryptographic Tools- Confidentiality with Symmetric Encryption (AES, DES, RC5, IDEA etc.), Asymmetric Encryption (RSA, DSA etc.), Message Authentication and Hash Functions, Public-Key Encryption, Digital Signatures and Key Management, Random and Pseudorandom Numbers, Practical Application: Encryption of Stored Data.

UNIT II. User Authentication- Means of Authentication, Password-Based Authentication, Token-Based Authentication, Biometric Authentication, Remote User Authentication, Security Issues for User Authentication, Practical Application: An Iris Biometric System, Case Study: Security Problems for ATM Systems.

UNIT III. Access Control- Access Control Principles, Subjects, Objects, and Access Rights, Discretionary Access Control, Example: UNIX File Access Control, RoleBased Access Control, Case Study: RBAC System for a Bank.

UNIT IV. Database Security-The Need for Database Security, Database Management Systems, Relational Databases, Database Access Control, Inference, Statistical Databases, Database Encryption, Cloud Security.

UNIT V. Malicious Software-Types of Malicious Software (Malware), Propagation–Infected Content–Viruses, Propagation–Vulnerability Exploit–Worms, Propagation–Social Engineering–SPAM E-mail, Trojans, Payload–System Corruption, Payload–Attack Agent–Zombie, Bots, Payload–Information Theft– Keyloggers, Phishing, Spyware, Payload–Stealthing–Backdoors, Rootkits,, Countermeasures.

UNIT VI. Denial-of-Service Attacks- Denial-of-Service Attacks, Flooding Attacks, Distributed Denial-of-Service Attacks, Application-Based Bandwidth Attacks, Reflector and Amplifier Attacks, Defenses Against Denial-of-Service Attacks, Responding to a Denial-of-Service Attack.

References/ Suggested Readings:

1. M. Stamp, Information Security: Principles and Practice, Wiley.

2. M. E. Whitman and H. J. Mattord, Principles of Information Security, Course Technology.
3. M. Bishop, Computer Security: Art and Science, Addison Wesley.
4. G. McGraw, Software Security: Building Security In, Addison Wesley

Course Name: Human Computer Interface

Course Code: BSCHCOSDSE505

Course Type: **DSE (Theoretical)**	Course Details: **DSEC-1 or DSEC-2**			L-T-P: **5 - 1 - 0**	
Credit: **6**	Full Marks: **50**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		….	**10**	….	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Provide an overview of the concepts relating to the design of human -computer interfaces in ways making computer-based systems comprehensive, friendly and usable.*
2. *Understand the theoretical dimensions of human factors involved in the acceptance of computer interfaces.*
3. *Understand the important aspects of implementation of human-computer interfaces.*
4. *Identify the various tools and techniques for interface analysis, design, and evaluation.*
5. *Identify the impact of usable interfaces in the acceptance and performance utilization of information systems.*

Course Content:

Theory

UNIT I. Introduction: Historical Evolution of HCI, Interactive System Design: Concept of Usability- Definition and Elaboration, HCI and Software Engineering, GUI Design and Aesthetics, Prototyping Techniques.

UNIT II. Model-Based Design and Evaluation: Basic Idea, Introduction to Different Types of Models, GOMS Family of Models (KLM And CMN -GOMS), Fitts' Law and Hickhyman's Law.

UNIT III. General Development Guidelines and Principles: Shneiderman's Eight Golden Rules, Norman's Seven Principles, Norman's Model of Interaction, Nielsen's Ten Heuristics with Example of its use, Contextual Inquiry.

UNIT IV. Dialog Design: Introduction to Formalism in Dialog Design, Design using FSM (Finite State Machines), State Charts and (Classical) Petri Nets in Dialog Design.

UNIT V. Task Modeling and Analysis: Hierarchical Task Analysis (HTA), Engineering Task Models and Concur Task Tree (CTT).

UNIT VI. Object Oriented Modeling: Object Oriented Principles, Definition of Class and Object and their Interactions, Object Oriented Modeling for User Interface Design, Case Study Related to Mobile Application Development.

References/ Suggested Readings:

1. A. Dix, J. Finlay, G. D. Abowd and R. Beale, Human Computer Interaction, Pearson Education.
2. J. Preece, Y. Rogers, H. Sharp, D. Baniyon, S. Holland and T. Carey, Human Computer Interaction, Addison-Wesley.
3. B.Shneiderman, Designing the User Interface, Addison Wesley.

Semester- VI

Course Name: Computer Graphics

Course Code: BSCHCOSC601

Course Type: **Core (Theoretical & Practical)**	Course Details: **CC-13**			L-T-P: **4 - 0 - 4**	
Credit: **6**	Full Marks: **100**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		30	**10**	**20**	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Acquire familiarity with the concepts and relevant mathematics of computer graphics.*
2. *Ability to implement various algorithms to scan, convert the basic geometrical primitives, transformations, area filling, clipping.*
3. *Describe the importance of viewing and projections.*
4. *Ability to design basic graphics application programs.*
5. *Familiarize with fundamentals of animation and Virtual reality technologies*
6. *Be able to design applications that display graphic images to given specifications.*
7. *To understand a typical graphics pipeline.*

Course Content:

Theory

UNIT I. Application Areas of Computer Graphics, Overview of Graphics Systems and Devices. Points and Lines, Line Drawing Algorithms, Mid-Point Circle and Ellipse Algorithms. Filled Area Primitives, Polygon Filling Algorithms. Curve Generation: Bezier and B-Spline Curves.

UNIT II. 2-D Geometrical Transforms: Translation, Scaling, Rotation, Reflection and Shear Transformations Composite Transforms, Transformations between Coordinate Systems. 2-D Viewing: The Viewing Pipeline, Viewing Coordinate Reference Frame, Window to Viewport Coordinate Transformation, Viewing Functions.

UNIT III. Line Clipping Algorithms- Cohen-Sutherland and Cyrus Beck Line Clipping Algorithms, Sutherland–Hodgeman Polygon Clipping Algorithm. 3-D Object Representation: Polygon Surfaces, Quadric Surfaces, Spline Representation

UNIT IV. 3-D Geometric Transformations: Translation, Rotation, Scaling, Reflection and Shear Transformations, Composite Transformations, 3-D Viewing: Viewing Pipeline, Viewing Coordinates, View Volume, General Projection Transforms and Clipping.

UNIT V. Visible Surface Detection Methods: Classification, Back -Face Detection, DepthBuffer, Scanline, Depth Sorting, BSP-Tree Methods, Area Sub-Division and Octree Methods Illumination Models and Surface Rendering Methods: Basic Illumination Models, Polygon Rendering Methods Computer Animation: Design of Animation Sequence, General Computer Animation Functions Key Frame Animation, Animation Sequence, Motion Control Methods, Morphing, Warping (Only Mesh Warping)

UNIT VI. Virtual Reality : Basic Concepts, Classical Components of VR System, Types of VR Systems, Three Dimensional Position Trackers, Navigation and Manipulation Interfaces, Gesture Interfaces. Input Devices, Graphical Rendering Pipeline, Haptic Rendering Pipeline, Open GL Rendering Pipeline. Applications of Virtual Reality.

Practical

The students are required to create interactive graphics applications in C using graphics application programming interfaces and demonstrate geometrical transformations. The lab material includes implementation of line drawings, circle drawing, ellipse drawing, line clipping, polygon clipping, polygon filling as well as different geometrical transformations.

Experiment 1: Line Drawing Using DDA and Bresenham.

Experiment 2: Circle Drawing Using Midpoint Algorithm.

Experiment 3: Ellipse Drawing Using Midpoint Algorithm.

Experiment 4: Curve Generation: Bezier and B-Spline Curves.

Experiment 5: Line Clipping Algorithms- Cohen-Sutherland and Cyrus Beck.

Experiment 6: Sutherland–Hodgeman Polygon Clipping Algorithm.

Experiment 7: Polygon Filling Algorithms.

Experiment 8: Performing the basic 2D transformations such as translation, Scaling, Rotation, shearing and reflection for a given 2D object.

Internal (CA) Evaluation: Practical Note Book (15 marks), Two Experiments (10 marks) – one from Experiment 1 to 4 and another from Experiment 5 to 8, Viva-voce (5 marks).

ESE Evaluation: Two Experiments (10 marks) - one from Experiment 1 to 4 and another from Experiment 5 to 8, Viva-voce (10 marks).

References/ Suggested Readings:

1. Donald Hearn and M. Pauline Baker, Computer Graphics with Open GL, Prentice Hall.
2. R. K Maurya, Computer Graphics with Virtual Reality, Wiley.
3. Foley, Van Dam, Feiner and Hughes, Computer Graphics Principles & practice, Pearson Education.
4. D. P. Mukherjee, Fundamentals of Computer Graphics and Multimedia, PHI.

Course Name: Machine Learning

Course Code: BSCHCOSC602

<table>
<tr><td>Course Type: Core (Theoretical & Practical)</td><td colspan="2">Course Details: CC-14</td><td colspan="2">L-T-P: 4 - 0 - 4</td></tr>
<tr><td rowspan="3">Credit: 6</td><td rowspan="3">Full Marks: 100</td><td colspan="2">CA Marks</td><td colspan="2">ESE Marks</td></tr>
<tr><td>Practical</td><td>Theoretical</td><td>Practical</td><td>Theoretical</td></tr>
<tr><td>30</td><td>10</td><td>20</td><td>40</td></tr>
</table>

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Differentiate between supervised, unsupervised machine learning approaches*
2. *Ability to choose appropriate machine learning algorithm for solving a problem*
3. *Design and adapt existing machine learning algorithms to suit applications*
4. *Understand the underlying mathematical relationships across various machine learning algorithms*
5. *Design and implement machine learning algorithms to real world applications*

Course Content:

Theory

UNIT I. Introduction: Concept of Machine Learning, Applications of Machine Learning, Key elements of Machine Learning, Supervised Learning, Classification, Unsupervised Learning, Clustering, Statistical Learning: Bayesian Method, The Naive Bayes Classifier.

UNIT II. Software's for Machine Learning and Linear Algebra Overview: Plotting of Data, Vectorization, Matrices and Vectors: Addition, Multiplication, Transpose and Inverse using Available Tool such as MATLAB.

UNIT III. Linear Regression: Prediction using Linear Regression, Gradient Descent, Linear Regression with one Variable, Linear Regression with Multiple Variables, Polynomial Regression, Feature Scaling/Selection.

UNIT IV. Logistic Regression: Classification using Logistic Regression, Logistic Regression vs. Linear Regression, Logistic Regression with one Variable and with Multiple Variables.

UNIT V. Regularization: Regularization and its Utility: The problem of Overfitting, Application of Regularization in Linear and Logistic Regression, Regularization and Bias/Variance.

UNIT VI. Neural Networks: Introduction, Model Representation, Gradient Descent vs. Perceptron Training, Stochastic Gradient Descent, Single Layer Perceptrons, Multilayer Perceptrons, Multiclass Representation, Back Propagation Algorithm, Radial Basis Function Networks.

Practical

Students are directed to do a minor project based on the contents of the course below for internal and ESE evaluation. Implement the minor project using Python/MATLAB and UCI machine learning repository [http://archive.ics.uci.edu/ml].

UNIT I. Implementation of different clustering algorithms – K-Means, Fuzzy c-means etc.

UNIT II. Implementation of The Naive Bayes Classifier.

UNIT III. Implementation of Linear Regression with one variable and multiple variables.

UNIT IV. Implementation of Logistic Regression with one variable and multiple variables.

UNIT V. Implementation of Gradient Descent learning.

UNIT VI. Implementation of Single Layer Perceptrons.

UNIT VII. Implementation of Multilayer Perceptrons Neural Networks with Back Propagation Algorithm.

UNIT VIII. Implementation of Radial Basis Function Neural Networks.

Internal (CA) Evaluation: Minor Project Report (15 marks), Demonstration of the minor project (10 marks), Viva-voce (5 marks).

ESE Evaluation: Presentation of the minor project (10 marks), Viva-voce (10 marks).

References/ Suggested Readings:

1. Ethem Alpaydin, Introduction to Machine Learning, The MIT Press.
2. Tom M. Mitchell, Machine Learning, Tata McGraw-Hill Education.
3. Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer.
4. Mevin P. Murphy, Machine Learning: A Probabilistic Perspective, The MIT Press.
5. Simon Haykin, Neural Networks and Learning Machines, Pearson.
6. D Dua and C Graff, UCI Machine Learning Repository [http://archive.ics.uci.edu/ml].

Course Name: Modelling and Simulation

Course Code: BSCHCOSDSE601

<table>
<tr><td>Course Type: DSE (Theoretical)</td><td colspan="2">Course Details: DSEC-3 or DSEC-4</td><td colspan="2">L-T-P: 5 - 1 - 0</td></tr>
<tr><td rowspan="3">Credit: 6</td><td rowspan="3">Full Marks: 50</td><td colspan="2">CA Marks</td><td colspan="2">ESE Marks</td></tr>
<tr><td>Practical</td><td>Theoretical</td><td>Practical</td><td>Theoretical</td></tr>
<tr><td>….</td><td>10</td><td>….</td><td>40</td></tr>
</table>

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Characterise systems in terms of their essential elements, purpose, parameters, constraints, performance requirements, sub-systems, interconnections and environmental context.*
2. *Understand the technical underpinning of modern computer simulation software.*
3. *System problem modelling and solving through the relationship between theoretical, mathematical, and computational modelling for predicting and optimizing performance and objective.*
4. *Mathematical modelling real world situations related to information systems development, prediction and evaluation of outcomes against design criteria.*
5. *Develop solutions and extract results from the information generated in the context of the information systems*
6. *Interpret the model and apply the results to resolve critical issues in a real world environment.*

Course Content:

Theory

UNIT I. Systems and environment: Concept of model and model building, model classification and representation, Use of simulation as a tool, steps in simulation study.

UNIT II. Continuous-time and Discrete-time systems: Laplace transform, transfer functions, statespace models, order of systems, z-transform, feedback systems, stability, observability, controllability. Statistical Models in Simulation: Common discrete and continuous distributions, Poisson process, empirical distributions.

UNIT III. Random Numbers: Properties of random numbers, generation of pseudo random numbers, techniques of random number generation, tests for randomness, random variate generation using inverse transformation, direct transformation, convolution method, acceptance-rejection.

UNIT IV. Design and Analysis of simulation experiments: Data collection, identifying distributions with data, parameter estimation, goodness of fit tests, selecting input models without data, multivariate an time series input models, verification and validation of models, static and dynamic simulation output analysis, steady -state simulation, terminating simulation, confidence interval estimation, Output analysis for steady state simulation, variance reduction techniques.

UNIT V. Queuing Models: Characteristics of queuing systems, notation, transient and steady state behavior, performance, network of queues.

UNIT VI. Large Scale systems: Model reduction, hierarchical control, decentralized control, structural properties of large-scale systems.

References/ Suggested Readings:

1. Stanislaw Raczynski, Modeling and Simulation: The Computer Science of Illusion, WILEY.

2. Hartmut Bossel, Modeling and Simulation, Springer.
3. Shailendra Jain, Modeling and Simulation using MATLAB - Simulink, Kindle edition.

Course Name: Theory of Computation

Course Code: BSCHCOSDSE602

Course Type: **DSE (Theoretical)**	Course Details: **DSEC-3 or DSEC-4**			L-T-P: **5 - 1 - 0**	
Credit: **6**	Full Marks: **50**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		….	**10**	….	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *To provide a formal connection between algorithmic problem solving and the theory of languages and automata and develop them into a mathematical (abstract) view towards algorithmic design and in general computation itself.*
2. *The course should in addition clarify the practical view towards the applications of these ideas in the engineering part as well.*
3. *Become proficient in key topics of theory of computation, and to have the opportunity to explore the current topics in this area.*

Course Content:

Theory

UNIT I. Automata: Introduction to Formal Proof, Additional Forms of Proof, Inductive Proofs, Finite Automata (FA), Deterministic Finite Automata (DFA), Non-Deterministic Finite Automata (NFA), Finite Automata with Epsilon Transitions.

UNIT II. Regular Expressions and Languages: Regular Expression, FA and Regular Expressions, Proving Languages not to be Regular, Closure Properties of Regular Languages, Equivalence and Minimization of Automata.

UNIT III. Context Free Grammars and Languages: Context Free Grammar (CFG), Parse Trees , Ambiguity in Grammars and Languages, Definition of The Pushdown Automata, Languages of a Pushdown Automata, Equivalence of Pushdown Automata and CFG Deterministic Pushdown Automata.

UNIT IV. Properties of Context Free Languages: Normal Forms for CFG, Pumping Lemma for CFL, Closure Properties of CFL, Turing Machines, Programming Techniques for TM, Variations of TM, Non Universal TM, Universal TM.

UNIT V. Undecidability: A Language that is not Recursively Enumerable (RE), an Undecidable Problem that is RE, Undecidable Problems about Turing Machine, Post's Correspondence Problem, The Classes P and NP.

References/ Suggested Readings:

1. J.E. Hopcroft, R. Motwani and J.D. Ullman, Introduction to Automata Theory, Languages and Computations, Pearson Education.
2. H.R. Lewis and C.H. Papadimitriou, Elements of the theory of Computation, Pearson Education.
3. K. L. P. Mishra and N. Chandrasekaran, Theory of Computer Science: Automata, Languages and Computation, PHI.
4. Thomas A. Sudkamp, An Introduction to the Theory of Computer Science, Languages and Machines, Pearson Education.
5. J. Martin, Introduction to Languages and the Theory of computation, Tata McGraw Hill.

Course Name: Data Mining

Course Code: BSCHCOSDSE603

Course Type: **DSE (Theoretical)**	Course Details: **DSEC-3 or DSEC-4**			L-T-P: **5 - 1 - 0**	
Credit: **6**	Full Marks: **50**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		….	**10**	….	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Demonstrate advanced knowledge of data mining concepts and techniques.*
2. *Apply the techniques of clustering, classification, association finding, feature selection and visualization on real world data.*
3. *Determine whether a real world problem has a data mining solution.*
4. *Apply data mining software and toolkits in a range of applications.*
5. *Set up a data mining process for an application, including data preparation, modelling and evaluation.*

Course Content:

Theory

UNIT I. Introduction to data mining (DM): Motivation for Data Mining - Data Mining-Definition and Functionalities – Classification of DM Systems - DM task primitives - Integration of a Data Mining system with a Database - Issues in DM – KDD Process

UNIT II. Data Pre-processing: Data summarization, data cleaning, data integration and transformation, data reduction, data discretization and concept hierarchy generation, feature extraction , feature transformation, feature selection, introduction to Dimensionality Reduction, CUR decomposition.

UNIT III. Concept Description, Mining Frequent Patterns, Associations and Correlations: What is concept description? - Data Generalization and summarization-based characterization - Attribute relevance - class comparisons, Basic concept, efficient and scalable frequent item-

set mining methods, mining various kind of association rules, from association mining to correlation analysis, Advanced Association Rule Techniques, Measuring the Quality of Rules.

UNIT IV. Classification and Prediction: Classification vs. prediction, Issues regarding classification and prediction, Statistical-Based Algorithms, Distance-Based Algorithms, Decision Tree Based Algorithms, Neural Network-Based Algorithms, Rule-Based Algorithms, Combining Techniques, accuracy and error measures, evaluation of the accuracy of a classifier or predictor. Neural Network Prediction methods: Linear and nonlinear regression, Logistic Regression Introduction of tools such as DB Miner / WEKA / DTREG DM Tools.

UNIT V. Cluster Analysis: Clustering: Problem Definition, Clustering Overview, Evaluation of Clustering Algorithms, Partitioning Clustering -K-Means Algorithm, KMeans Additional issues, PAM Algorithm; Hierarchical Clustering – Agglomerative Methods and divisive methods, Basic Agglomerative Hierarchical Clustering, Strengths and Weakness; Outlier Detection, Clustering high dimensional data, clustering Graph and Network data.

UNIT VI. Web mining and other data mining: Web Mining: Introduction to Web Mining, Web content mining, Web usage mining, Web Structure mining, Web log structure and issues regarding web logs, Spatial Data Mining, Temporal Mining, And Multimedia Mining. Applications of Distributed and parallel Data Mining.

UNIT VII. ROC Analysis, Data Mining Trends, Big Data, Data Analytics.

References/ Suggested Readings:

1. Arun K Pujari, Data Mining Techniques, Universities Press.
2. J. Han, M. Kamber, Data Mining Concepts and Techniques, Morgan Kaufmann.
3. M. Kantardzic, Data mining: Concepts, models, methods and algorithms, John Wiley & Sons Inc.
4. M. Dunham, Data Mining: Introductory and Advanced Topics, Pearson Education.
5. Ning Tan, Vipin Kumar, Michael Steinbanch Pang, Introduction to Data Mining, Pearson Education.

Course Name: Cloud Computing

Course Code: BSCHCOSDSE604

<table>
<tr><td colspan="2">Course Type: DSE (Theoretical)</td><td colspan="2">Course Details: DSEC-3 or DSEC-4</td><td colspan="2">L-T-P: 5 - 1 - 0</td></tr>
<tr><td rowspan="3">Credit: 6</td><td rowspan="3">Full Marks: 50</td><td colspan="2">CA Marks</td><td colspan="2">ESE Marks</td></tr>
<tr><td>Practical</td><td>Theoretical</td><td>Practical</td><td>Theoretical</td></tr>
<tr><td>….</td><td>10</td><td>….</td><td>40</td></tr>
</table>

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Analyze the trade-offs between deploying applications in the cloud and over the local infrastructure.*

2. *Compare the advantages and disadvantages of various cloud computing platforms.*
3. *Deploy applications over commercial cloud computing infrastructures such as Amazon Web Services, Windows Azure, and Google AppEngine.*
4. *Program data intensive parallel applications in the cloud.*
5. *Analyze the performance, scalability, and availability of the underlying cloud technologies and software.*
6. *Identify security and privacy issues in cloud computing.*
7. *Explain recent research results in cloud computing and identify their pros and cons.*
8. *Solve a real-world problem using cloud computing through group collaboration.*

Course Content:

Theory

UNIT I. Introduction to cloud computing: Definition, characteristics, components, Cloud service provider, the role of networks in Cloud computing, Cloud deployment models- private, public & hybrid, Cloud service models, multitenancy, Cloud economics and benefits, Cloud computing platforms - IaaS: Amazon EC2, PaaS: Google App Engine, Microsoft Azure, SaaS.

UNIT II. Virtualization: Virtualization concepts , Server virtualization, Storage virtualization, Storage services, Network virtualization, Service virtualization, Virtualization management, Virtualization technologies and architectures, virtual machine, Measurement and profiling of virtualized applications. Hypervisors: KVM, Xen, VMware hypervisors and their features.

UNIT III. Data in cloud computing: Relational databases, Cloud file systems: GFS and HDFS, BigTable, HBase and Dynamo. MapReduce and extensions: Parallel computing, the map-Reduce model, Parallel efficiency of MapReduce, Relational operations using Map-Reduce, Enterprise batch processing using MapReduce.

UNIT IV. Cloud security: Cloud security fundamentals, Vulnerability assessment tool for cloud, Privacy and Security in cloud. Cloud computing security architecture: General Issues, Trusted Cloud computing, Secure Execution Environments and Communications, Micro - architectures; Identity Management and Access control, Autonomic security, Security challenges : Virtualization security management - virtual threats, VM Security Recommendations, VM - Specific Security techniques, Secure Execution Environments and Communications in cloud.

UNIT V. Issues in cloud computing: Implementing real time application over cloud platform, Issues in Inter-cloud environments, QOS Issues in Cloud, Dependability, data migration, streaming in Cloud. Quality of Service (QoS) monitoring in a Cloud computing environment. Cloud Middleware. Mobile Cloud Computing. Inter Cloud issues. A grid of clouds, Sky computing, load balancing, resource optimization, resource dynamic reconfiguration, Monitoring in Cloud.

References/ Suggested Readings:

1. Gautam Shroff, Enterprise Cloud Computing,Cambridge publication
2. Ronald Krutz and Russell Dean Vines, Cloud Security, Wiley-India.
3. Dr. Kumar Saurabh, Cloud Computing, Wiley Publication.

Course Name: Internet of Things

Course Code: BSCHCOSDSE605

Course Type: **DSE (Theoretical)**	Course Details: **DSEC-3 or DSEC-4**			L-T-P: **5 - 1 - 0**	
Credit: **6**	Full Marks: **50**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		….	**10**	….	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *To learn the concepts of Sensors, Wireless Network and Internet*
2. *To learn and implement use of Devices in IoT technology.*
3. *To learn the different IoT Technologies like Micro-controller, Wireless communication like Blue Tooth, GPRS, Wi-Fi and Storage and embedded systems*
4. *To understand how to program on embedded and mobile platforms including different Microcontrollers like ESP8266, Raspberry Pi, Arduino and Android programming*
5. *To understand how to make sensor data available on the Internet (data acquisition) and understand how to analyze and visualize sensor data*
6. *To understand, analysis and evaluate different protocols used in IoT.*
7. *To learn basic python programming for IoT applications*
8. *To learn and design different applications in IoT.*
9. *To design, develop and test different prototypes in IoT.*

Course Content:

Theory

UNIT I. Introduction to IoT, Sensors and Actuators: Introduction to IoT: Definition, Characteristics, Applications, Evolution, Enablers, Connectivity Layers, Addressing, Networking and Connectivity Issues, Network Configurations, Multi-Homing, Sensing: Sensors and Transducers, Classification, Different Types of Sensors, Errors, Actuation: Basics, Actuator Types- Electrical, Mechanical Soft Actuators.

UNIT II. Introduction to Networking, Communication Protocols and Machine-to-Machine Communication: Basics of Networking, Communication Protocols, Sensor Network, Machine to Machine Communication (IoT Components, Inter-Dependencies, SoA, Gateways, Comparison Between IoT & Web, Difference Protocols, Complexity of Networks, Wireless Networks, Scalability, Protocol Classification, MQTT & SMQTT, IEEE 802.15.4, Zigbee).

UNIT III. Arduino Programming: Interoperability in IoT, Introduction To Arduino Programming, Integration Of Sensors And Actuators With Arduino.

UNIT IV. Python Programming and Raspberry Pi: Introduction to Python Programming, Introduction to Raspberry Pi, Implementation of IoT with Raspberry Pi, Implementation of IoT with Raspberry Pi.

UNIT V. (Data Analytics and Cloud Computing) Data Handling and Analytics, Cloud Computing Fundamentals, Cloud Computing Service Model, Cloud Computing Service Management and Security, Sensor-Cloud Architecture, View and Dataflow.

UNIT VI. (FOG Computing and Case Studies) FOG Computing: Introduction, Architecture, Need, Applications and Challenges.

UNIT VII. Industrial IoT, Case Studies: Agriculture, Healthcare, Activity Monitoring.

References/ Suggested Readings:

1. Pethuru Raj and Anupama C. Raman, The Internet of Things: Enabling Technologies, Platforms, and Use Cases, CRC Press.
2. A Bahga and Vijay Madisetti, Internet of Things: A Hands-on Approach, Universities Press.
3. Samuel Greengard, The Internet of Things, Cambridge, MA: MIT Press.