Third Semester (CSE)

Course Code	Type of Course	Course Title	Hours per Week			Marks Distribution		Total Marks	Credits
			L	T	P	Internal	External		
BTCS-18301	Professional Core Course	Digital Electronics	3	0	0	40	60	100	3
BTCS-18302	Professional Core Course	Data structure & Algorithms	3	0	0	40	60	100	3
BTCS-18303	Engineering Science Course	Object Oriented Programming	3	0	0	40	60	100	3
BTAM-18301	Basic Science Course	Mathematics-III	3	1	0	40	60	100	4
BTHS-18901	Humanities & Social Sciences Including Management Course	Fundamentals of Management for Engineers	3	0	0	40	60	100	3
BTCS-18304	Professional Core Course	Digital Electronics Lab	0	0	2	30	20	50	1
BTCS-18305	Professional Core Course	Data structure & Algorithms Lab	0	0	2	30	20	50	1
BTCS-18306	Engineering Science Course	Object Oriented Programming lab.	0	0	2	30	20	50	1
BTCS-18307	Professional Core Course	IT Workshop**	1	0	2	30	20	50	2
BTCS-18308	Internship	Institutional Summer Training*				100	0	100	3
Total			**16**	**0**	**8**	**420**	**380**	**800**	**24**

*This training will be performed by the students in the college in CSE lab for learning Python etc / Central workshop.

** only Practical Examination will be held. No theory Examination is to be held

Fourth Semester(CSE)

Course Code	Type of Course	Course Title	Hours per Week			Marks Distribution		Total Marks	Credits
			L	T	P	Internal	External		
BTAM-18401	Professional Core Course	Discrete Structures	3	1	0	40	60	100	4
BTCS-18402	Engineering Science Course	Computer Organization & Architecture	3	0	0	40	60	100	3
BTCS-18403	Professional Core Course	Operating Systems	3	0	0	40	60	100	3
BTCS-18404	Professional Core Course	Design & Analysis of Algorithms	3	0	0	40	60	100	3
BTHS-18904	Humanities & Social Sciences including Management Course	Organizational Behaviour	3	0	0	40	60	100	3
CSMC-I	Mandatory Course	Environmental Sciences*	-	-	-	-	-	-	0
BTCS-18405	Engineering Science Course	Computer Organization & Architecture Lab	0	0	2	30	20	50	1
BTCS-18406	Professional Core Course	Operating Systems Lab	0	0	2	30	20	50	1
BTCS-18407	Professional Core Course	Design & Analysis of Algorithms Lab	0	0	2	30	20	50	1
Total			**15**	**1**	**6**	**290**	**360**	**650**	**19**

*Students will be given topics related to this course for presentation in the department.

Fifth Semester(CSE)

Course Code	Type of Course	Course Title	Hours per Week			Marks Distribution		Total Marks	Credits
			L	T	P	Internal	External		
BTCS-18501	Engineering Science Course Course	Programming in JAVA	3	0	0	40	60	100	3
BTCS-18502	Professional Core Course	Database Management Systems	3	0	0	40	60	100	3
BTCS-18503	Professional Core Course	Formal Language & Automata Theory	3	0	0	40	60	100	3
BTCS-18504	Professional Core Course	Computer Networks	3	0	0	40	60	100	3
BTHS-18905	Humanities & Social Sciences including Management Course	Effective Technical Communication	3	0	0	40	60	100	3
BTCS-18xxx	Professional Elective	Elective-I	3	0	0	40	60	100	3
CSMC-II	Mandatory Course	Constitution of India/ Essence of Indian Traditional Knowledge*	-	-	-	-	-	-	0
BTCS-18505	Professional Core Course	Database Management Systems Lab	0	0	2	30	20	50	1
BTCS-18506	Professional Core Course	Computer Networks Lab	0	0	2	30	20	50	1
BTCS-18507	Engineering Science	Programming in Java Lab	0	0	2	30	20	50	1
BTCS-18508	Internship	Summer internship					100	100	3
Total			**18**	**0**	**6**	**330**	**520**	**850**	**24**

**Students will be given topics related to this course for presentation in the department.

Sixth Semester(CSE)

Course Code	Type of Course	Course Title	Hours per Week			Marks Distribution		Total Marks	Credits
			L	T	P	Internal	External		
BTCS-18601	Professional Core Course	Compiler Design	3	0	0	40	60	100	3
BTCS-18602	Professional Core Course	Software Engineering	3	0	0	40	60	100	3
BTCS-18xxx	Professional Elective Course	Elective-II	3	0	0	40	60	100	3
BTCS-18xxx	Professional Elective Course	Elective-III	3	0	0	40	60	100	3
BTxx-18xxx	Open Elective Courses	Open Elective-I (Humanities)	3	0	0	40	60	100	3
BTCS-18603	Project	Minor Project	0	0	4	60	40	100	2
BTCS-18604	Professional Core Course	Compiler Design Lab	0	0	2	30	20	50	1
BTCS-18605	Professional Core Course	Software Engineering Lab	0	0	2	30	20	50	1
Total			**15**	**0**	**14**	**320**	**380**	**700**	**19**

Seventh Semester(CSE)

Course Code	Type of Course	Course Title	Hours per Week			Marks Distribution		Total Marks	Credits
			L	T	P	Internal	External		
BTCS-18xxx	Professional Elective Course	Elective-IV	3	0	0	40	60	100	3
BTCS-18xxx	Professional Elective Course Courses	Elective-V	3	0	0	40	60	100	3
BTxx-18xxx	Open Elective Course	Open Elective-II	3	0	0	40	60	100	3
BTCS-18701	Professional Core Course	Artificial Intelligence	3	0	0	40	60	100	3
BTCS-18702	Project	Major Project-I	0	0	8	100	50	150	4
BTCS-18703	Professional Core Course	Artificial Intelligence Lab	0	0	2	30	20	50	1
BTCS-18704	Internship	Summer industrial Training						100	6
Total			**14**	**0**	**10**	**340**	**310**	**750**	**23**

Eighth Semester(CSE)

Course Code	Type of Course	Course Title	Hours per Week			Marks Distribution		Total Marks	Credits
			L	T	P	Internal	External		
BTCS-18xxx	Professional Elective Course	Elective-VI	3	0	0	40	60	100	3
BTxx-18xxx	Open Elective Course	Open Elective-III	3	0	0	40	60	100	3
BTCS-18801	Professional Core Course	Cloud Computing	3	0	0	40	60	100	3
BTCS-18802	Project	Major Project-II	0	0	8	100	50	150	4
BTCS-18803	Internship	*Seminar	2	-	-	50	0	50	2
Total			**11**	**0**	**12**	**270**	**230**	**500**	**15**

***The students will prepare and present seminar based on their Project

LIST OF ELECTIVES

Elective-I

BTCS-18510 Signal & Systems
BTC-18511 Web Technologies
BTCS-18512 Computational Biology
BTCS-18513 Computer Graphics

Elective-II

BTCS-18610 Mobile Application Development
BTCS-18611 Distributed Systems
BTCS-18612 Machine Learning
BTCS-18613 Digital Signal Processing
BTCS-18614 Open Source Technologies

Elective-III

BTCS-18615 Parallel and Distributed Algorithms
BTCS-18616 Embedded Systems
BTCS-18617 Microprocessor and Assembly Language Programming
BTCS-18718 Ad-Hoc and Sensor Networks

Elective-IV

BTCS-18710 Information Theory and Coding
BTCS-18711 Distributed Operating System
BTCS-18712 Soft Computing
BTCS-18713 Human Computer Interaction

Elective-V

BTCS-18715 Computational Number Theory
BTCS-18716 Speech and Natural Language Processing
BTCS-18717 Parallel Architectures
BTCS-18718 Data Mining

Elective-VI

BTCS-18810 Queuing Theory and Modeling
BTCS-18811 Real Time Systems
BTCS-18812 Data Analytics
BTCS-18813 Image Processing

LIST OF OPEN ELECTIVES

Open electives offered by the department:

Courses of odd semesters:

BTCS-18302 Data Structures & Algorithms

BTCS-18303 Object Oriented Programming

BTCS-18504 Computer Networks

Courses of even semesters:

BTCS-18402 Computer organization & Architecture

BTCS-18403 Operating system

BTCS-18502 Database Management System

LIST OF COURSES FOR HONOURS DEGREE

In order to have an Honours degree, a student chooses 18-20 credits from the following courses in addition.

Course Code	Type of courses	Course Title	Hours per Week			Marks Distributions		Total Marks	Credits
			L	T	P	Internal	External		
BTCS-H-18101	Professional Elective Courses	Graph Theory	3	0	0	40	60	100	3
BTCS-H-18102	Professional Elective Courses	Computer Vision	3	0	0	40	60	100	3
BTCS-H-18103	Professional Elective Courses	Embedded Systems	3	0	0	40	60	100	3
BTCS-H-18104	Professional Elective Courses	Software Project Management	3	0	0	40	60	100	3
BTCS-H-18105	Professional Elective Courses	Cryptography & Network Security	3	0	0	40	60	100	3
BTCS-H-18106	Professional Elective Courses	Internet-of-Things	3	0	0	40	60	100	3
BTCS-H-18107	Professional Elective Courses	Data Analytics	3	0	0	40	60	100	3
BTCS-H-18108	Professional Elective Courses	Machine Learning	3	0	0	40	60	100	3
BTCS-H-18109	Professional Elective Courses	ICT in Agriculture and Rural Development	3	0	0	40	60	100	3
BTCS-H-18110	Professional Elective Courses	Computational Technologies for Smart Cities	3	0	0	40	60	100	3
BTCS-H-18111	Professional Elective Courses	Computer Forensics	3	0	0	40	60	100	3
BTCS-H-18112	Professional Elective Courses	Advanced Algorithms	3	0	0	40	60	100	3

BTCS-18301
Digital Electronics **L T P**
3 0 0

Internal Marks: 40
External Marks: 60
Total Marks: 100

Objective/s and Expected outcome:
Demonstrate the operation of simple digital gates, identify the symbols, develop the truth table for those gates; combine simple gates into more complex circuits; change binary, hexadecimal, octal numbers to their decimal equivalent and vice versa, demonstrate the operation of a flip-flop. Design counters and clear the concept of shift resisters. Study different types of memories and their applications. Convert digital into analog and vice versa.

Number Systems:
Binary, Octal, Decimal, Hexadecimal. Number base conversions, 1's, 2's, complements, signed Binary numbers. Binary Arithmetic, Binary codes: Weighted BCD, Gray code, Excess 3 code, ASCII, conversion from one code to another.

Boolean Algebra:
Boolean postulates and laws–De-Morgan'sTheorem, Principle of Duality, Boolean expression – Boolean function, Minimization of Boolean expressions – Sum of Products (SOP), Product of Sums (POS), Min term, Max term, Canonical forms, Conversion between canonical forms, Karnaugh map Minimization, Quine-McCluskey method - Don't care conditions.

Logic GATES:
AND, OR, NOT, NAND, NOR, Exclusive-OR and Exclusive-NOR. Implementations of Logic Functions using gates, NAND-NOR implementations. Study of logic families like RTL, DTL, DCTL, TTL, MOS, CMOS, ECL and their characteristics.

Combinational Circuits:
Design procedure–Adders, Subtractors, Serial adder/Subtractor, Parallel adder/Subtractor Carry look ahead adder, BCD adder, Magnitude Comparator, Multiplexer/Demultiplexer, encoder/decoder, parity checker, code converters. Implementation of combinational logic using MUX.

Sequential Circuits:
Flip flops SR, JK, T, D and Master slave, Excitation table, Edge triggering, Level Triggering, Realization of one flip flop using other flip flops. Asynchronous/Ripple counters, Synchronous counters, Modulo-n counter, Ring Counters, Shift registers.

Memory Devices:
Classification of memories, RAM organization, Write operation, Read operation, Memory cycle. Static RAM Cell, MOSFET RAM cell, Dynamic RAM cell. ROM organization, PROM, EPROM, EEPROM, Field Programmable Gate Arrays (FPGA).

Signal Conversions: Analog & Digital signals. Introduction to A/D and D/A conversion techniques

Suggested Readings/ Books:

1 Morris Mano, Digital Design, Prentice Hall of India Pvt. Ltd
2 Donald P.Leach and Albert Paul Malvino, Digital Principles and Applications, Tata McGraw Hill Publishing Company Limited, New Delhi,

COURSE OUTCOMES (CO): The student is expected to

1. Solve basic binary math operations using the logic gates.
2. Demonstrate programming proficiency using the various logical elements to design practically motivated logical units.
3. Design different units that are elements of typical computer's CPU.
4. Apply knowledge of the logic design course to solve problems of designing of control units of different input/output devices.

BTCS-18302

Data Structure and algorithms

L T P
3 0 0

Internal Marks: 40
External Marks: 60
Total Marks: 100

Objectives of the course:

1 To impart the basic concepts of data structures and algorithms.
2 To understand concepts about searching and sorting techniques
3 To understand basic concepts about stacks, queues, lists trees and graphs.
4 To enable them to write algorithms for solving problems with the help of fundamental Data structures

Detailed contents:

Module1:
Introduction: Basic Terminologies: Elementary Data Organizations, Data Structure Operations: insertion, deletion, traversal etc.; Analysis of an Algorithm, Asymptotic Notations, Time-Space trade off. **Searching:** Linear Search and Binary Search Techniques and their complexity analysis.

Module2:
Stacks and Queues: ADT Stack and its operations: Algorithms and their complexity analysis, Applications of Stacks: Expression Conversion and evaluation – corresponding algorithms and complexity analysis. ADT queue, Types of Queue: Simple Queue, Circular Queue, Priority Queue; Operations on each types of Queues: Algorithms and their analysis.

Module3:
Linked Lists: Singly linked lists: Representation in memory, Algorithms of several operations: Traversing, Searching, Insertion into, Deletion from linked list; Linked representation of Stack and Queue, Header nodes, Doubly linked list: operations on it and algorithmic analysis; Circular Linked Lists: all operations their algorithms and the complexity analysis.

Trees: Basic Tree Terminologies, Different types of Trees: Binary Tree, Threaded Binary Tree, Binary Search Tree, AVL Tree; Tree operations on each of the trees and their algorithms with complexity analysis.

Module4:
Sorting and Hashing: Objective and properties of different sorting algorithms: Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort; Performance and Comparison among all the methods, Hashing.
Graph: Basic Terminologies and Representations, Graph search and traversal algorithms . Minimum Spanning Trees : Prim's algorithm, Kruskul's algorithm

Suggested books:

1. "Fundamentals of Data Structures", Illustrated Edition by Ellis Horowitz, Sartaj Sahni, Computer Science Press.

Suggested reference books:

1 Algorithms, Data Structures, and Problem Solving with C++", Illustrated Edition by Mark Allen Weiss, Addison-Wesley Publishing Company
2 "How to Solve it by Computer", 2nd Impression by R. G. Dromey, Pearson Education.

Course outcomes

1. For a given algorithm student will able to analyze the algorithms to determine the time and computation complexity and justify the correctness.

1 For a given Search problem (Linear Search and Binary Search) student will able to implement it.
2 For a given problem of Stacks, Queues and linked list student will able to implement it and analyze the same to determine the time and computation complexity.
3 Student will able to write an algorithm Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort and compare their performance in term of Space and Time complexity.
4 Student will able to implement Graph search and traversal algorithms and determine the time and computation complexity.

BTCS-18303
Object Oriented Programming

Objectives of the course
The course will introduce standard tools and techniques for software development, using object oriented approach. To understand Object Oriented Programming concepts and basic characteristics of C++.

Introduction
What is object oriented programming? Procedural Vs. Object-Oriented Programming , Basic Concepts and Principles of OOP

C++ Programming basics
Overview of C++, Program Structure, Exploring the Basic Components of C++ , Type Casting in C++, Operators in C++, Control Structures

Functions
Explore Functions , Describing Call by Value and Call by Reference , Inline Function, Overloading of Functions, String Library Functions, Recursive Functions, Friend Function.

Objects and Classes
Basics of Object and Class, Private and Public Members, Member Functions, Scope Resolution Operator, Constructors and their types, Destructors, Passing Objects as Function Parameters, Returning Objects from Functions.

Inheritance
Concept of inheritance, Derived class and base class, Types of Inheritance, Ambiguity and solution while implementing Multiple Inheritance.

Polymorphism
Concept of Polymorphism, Types of polymorphism, Function Overloading, Operator Overloading, Function Overriding.

Memory Management
Introduction to Pointers, Pointers and Objects, Dynamic Memory Management using new and delete operators, The this Pointer, pointer to object.

Templates and Exception Handling
Concept of Generic Programming, Function Template, Class Template, Exception handling mechanism, use of try, catch and throw keywords

Streams and Files
File Stream Operations, Opening and Closing a File, File Modes, File Pointers, Input Output Operations, Reading/Writing an object into file.

The concepts should be practiced using C++.

Suggested books
1. Lafore R., Object Oriented Programming in C++, Waite Group
2. E. Balagurusamy, Object Oriented Programming with C++, Tata McGraw Hill

Course Outcomes

After taking the course, students will be able to:

1. Recognise features of object-oriented design such as encapsulation, polymorphism, inheritance, and composition of systems based on object identity.
2. Apply these features to program design and implementation.
3. Design applications by using these object oriented concepts.

BTAM-18301

Mathematics – III

3L: 1T: 0P

Objectives

The objective of this course is to familiarize the students with functions of several variables and to introduce effective mathematical tools for the solutions of ordinary differential equations.

Multivariable Calculus (Differentiation) 12 hours

Partial differentiation, total derivative, Composite functions, Implicit functions, Euler's Theorem, Jacobians. Applications: Tangent plane and normal line, Taylor's and Maclaurin Theorem, Application in estimation of error and approximation, Maxima and Minima, Method of Lagrange's multipliers.

Multivariable Calculus (Integration) 10 hours

Double Integral, Change of variable in double integral, Change of order of Integration, Triple Integral, Change of variable in Triple Integral, Applications of Double and Triple Integrals

Ordinary Differential Equations 14 hours

Brief review of first order ordinary differential equations, Exact equations, Equations reducible to exact equations, Equations of the first order and higher degree, Clairaut's equation. Linear differential equations with constant co-efficient, Complimentary functions and particular integral, Method of variation of parameters, Equations reducible to linear equations with constant co-efficient (Cauchy's and Legendre's linear equations), Simultaneous linear equations with constant co-efficient.

Course outcomes

The students will be able to:

- apply mathematical tools of differentiation and integration of functions of multiple variables which are used in various techniques dealing engineering problems.
- understand the methods which can be used to solve the first and higher order ordinary differential equations.

Textbooks/References:

1. Erwin Kreyszig, Advanced Engineering Mathematics, John Wiley & Sons.
2. G.B. Thomas and R.L. Finney, Calculus and Analytic geometry, Pearson.
3. Ramana B.V., Higher Engineering Mathematics, Tata McGraw Hill New Delhi.
4. B.S. Grewal, Higher Engineering Mathematics, Khanna Publishers.
5. S. L. Ross, Differential Equations, Wiley India
6. T. Veerarajan, Engineering Mathematics for first year, Tata McGraw-Hill, New Delhi.

BTHS-18901

Fundamentals of Management for Engineers

L T P
3 0 0

Internal Marks: 40
External Marks: 60
Total Marks: 100

Unit 1. Definition, functions, and significance of Management. Levels of management, Douglas Mc Gregor Theory X and Theory Y. Difference between management and Administration.

Unit 2. Evolution of Management, thought, approaches of management. principles of Henry Fayol and F.W Taylor.

Unit3. Planning and organization nature, objectives and significance of planning, types and steps of planning. Span of control. Methods and types of training, Various organizational structures. Formal and informal organizations.

Unit 4. Concept of motivation, theories of motivation - Maslow need hierarchy theory & Herzberg two factor theory, Concepts of leadership and styles. Steps of Controlling .

Books Recommended:-

1. General Management - C.B. Gupta Sultan Chand

2. Principal and Practice of management- L.M. Prasad Sultan Chand

3. Essential of Management -Koontz & O, Donnel Tata Mc Graw

4. Essential Of Management – Koontz and Weihrich Tata Mc Graw 5. Management : James Stoner, R Edward Freeman, Daniel R. Gilbert, Jr. Prentice Hall of India

BTCS-18304
Digital Electronics Lab

Internal Marks: 30
External Marks: 20

OBJECTIVES : At the end, students should be able to implement Logic Gates: Truth-table verification of OR, AND, NOT, XOR, NAND and NOR gates; Realization of OR, AND, NOT and XOR functions using universal gates, Half Adder / Full Adder, Half Subtractor / Full Subtractor, 4-Bit Binary-to-Gray & Gray-to-Binary Code Converter, 4-Bit and 8-Bit Comparator, Multiplexer, Demultiplexer, and Flip Flops.

Implementation all experiments with help of Bread- Board.

Implementation all experiments with help of Bread- Board.

1. Study of Logic Gates: Truth-table verification of OR, AND, NOT, XOR, NAND and NOR gates;Realization of OR, AND, NOT and XOR functions using universal gates.
2. Half Subtractor / Full Subtractor: Realization using basic XOR gates.
3. Half Subtractor / Full Subtractor: Realization using NAND gates.
4. 4-Bit Binary-to-Gray & Gray-to-Binary Code Converter: Realization using XOR gates.
5. 4-Bit and 8-Bit Comparator: Implementation using IC7485 magnitude comparator chips.
6. Multiplexer: Truth-table verification and realization of Half adder and Full adder using IC74153 chip.
7. Demultiplexer: Truth-table verification and realization of Half subtractor and Full subtractor using IC74139 chip.
8. Flip Flops: Truth-table verification of JK Master Slave FF, T-type and D-type FF using IC7476 chip.
9. Asynchronous Counter: Realization of 4-bit up counter and Mod-N counter using IC7490 & IC7493 chip.
10. Synchronous Counter: Realization of 4-bit up/down counter and Mod-N counter using IC74192 & IC74193 chip.
11. Shift Register: Study of shift right, SIPO, SISO, PIPO, PISO & Shift left operations using IC7495 chip.
12. DAC Operation: Study of 8-bit DAC (IC 08/0800 chip), obtain staircase waveform using IC7493 chip.
13. ADC Operations: Study of 8-bit ADC.

COURSE OUTCOMES (CO): The student is expected to:

1 Operate laboratory equipment.
2 Construct, analyze, and troubleshoot simple combinational and sequential circuits.
3 Design and troubleshoot a simple state machine.
4. Measure and record the experimental data, analyze the results, and prepare a formal laboratory report.

BTCS-18305

Data Structures Lab

Internal Marks: 30 **L T P**
External Marks: 20 **0 0 2**
Total Marks: 50

List of Practical Exercises, to be implemented using object-oriented approach in C++ Language

1. Write a menu driven program that implements following operations (using separate functions) on a linear array
   - Insert a new element at end as well as at a given position
   - Delete an element from a given array whose value is given or whose position is given
   - To find the location of a given element
   - To display the elements of the linear array
2. Write a menu driven program that maintains a linear linked list and implements the following operations (using separate functions):
   - Insert a new element
   - Delete an existing element
   - Search an element
   - Display all the elements
3. Write a program to demonstrate the use of stack (implemented using linear array) in converting arithmetic expression from infix notation to postfix notation.
4. Program to demonstrate the use of stack (implemented using linear linked lists) in evaluating arithmetic expression in postfix notation.
5. Program to demonstration the implementation of various operations on a linear queue represented using a linear array.
6. Program to demonstration the implementation of various operations on a circular queue represented using a linear array.
7. Program to demonstration the implementation of various operations on a queue represented using a linear linked list (linked queue).
8. Program to illustrate the implementation of different operations on a binary search tree.
9. Program to illustrate the implementation of graph using adjacency matrix.
10. Program to sort an array of integers in ascending order using bubble sort.
11. Program to sort an array of integers in ascending order using selection sort.
12. Program to sort an array of integers in ascending order using insertion sort.
13. Program to sort an array of integers in ascending order using quick sort.
14. Program to demonstrate the use of binary search to search a given element in a sorted array in ascending order.

COURSE OUTCOMES (CO):

The student is expected to:

1. To design and analyze the time and space efficiency of the data structure
2. Identity the appropriate data structure for given problem
3. To have practical knowledge on the application of data structures

BTCS-18306
Object Oriented Programming Lab

Internal Marks: 20
External Marks: 30
Total Marks: 50

1. Introduction to OOP lab (Simple C++ program)
2. WAP to demonstrate the use of Classes and Objects
3. Constructors and Destructors; Write a program to demonstrate different types of constructors and destructors.
4. Operator overloading; Write a program for overloading various unary operators
5. Write a program for overloading various binary operators
6. Memory Management; Write a program to demonstrate the use of new and delete keywords
7. Inheritance; Write a program to demonstrate different types of inheritance
8. Write a program to remove ambiguity from hybrid inheritance
9. Polymorphism; Write a program for polymorphism(virtual function)
10. Write a program for templates (class and function template)
11. File handling; Write a program to copy contents of one file to another file.
12. Program using streams

Course outcomes:

The student is expected to:
1. Conceptualize the given problem and transform it in to an Object Oriented system.
2. Implement coding standard and verification practices
3. Build expertise in Object Oriented programming language.

BTCS-18307
IT Workshop

L T P
1 0 2

Objective of Course: The objective of this course is to demonstrate the students, the basic features of MATLAB/SCILAB and hands on training on any of them so that the students could be able to use this tool later on for various projects and thesis work.

Basics of MATLAB/SCILAB : Introduction, Basic features, A minimum session, Starting MATLAB/Scilab, using MATLAB/Scilab as a calculator , Quitting MATLAB/SCILAB , creating variables, Overwriting variable, Error messages, Making corrections, Controlling the hierarchy of operations or precedence , Controlling the appearance of floating point number , Managing the workspace, Entering multiple statements per line.

Basic Graphics and Matrix Algebra: Mathematical functions, Basic plotting, overview, Creating simple plots , Adding titles, axis labels, and annotations , multiple data sets in one plot, Specifying line styles and colors, Entering a vector, Entering a matrix, Matrix indexing, Colon operator, Linear spacing, Colon operator in a matrix, creating a sub-matrix, Deleting row or column, Transposing a matrix , Concatenating matrices.

Array operations and Linear equations: Array operations, Matrix arithmetic operations , Array arithmetic operations , Solving linear equations , Matrix inverse

Introduction to programming in MATLAB/SCILAB : Introduction , M-File Scripts, Examples, Script side-effects , File functions , Anatomy of a M-File function, Input and output arguments , Input to a script file , Output commands

Control flow and operators: Introduction , Control flow, The ‘‘if...end’’ structure, Relational and logical operators , The‘‘for...end’’loop , The ‘‘while...end’’loop Other flow structures, Saving output to a file.

List of Experiments: (using MATLAB or SCILAB)

1. Installation of MATLAB/SCILAB.
2. Various Operations of mathematics in MATLAB/SCILAB.
3. Handling of matrices.
4. To plot different types of two dimensional plots.
5. Branching statments-IF,IF-ELSE,Switch.
6. Write a program for printing below mention pattern.

```
1
1  2
1  2  3
1  2  3  4
```

7. Write a program for calculating whether number is prime or not with the help of function.
8. Write a program for implementation of basic Calculator.

Text Book:

1. Stephen J. Chapman . MATLAB- Programming for Engineers, Fourth Edition 2008
2. Holly More.MATLAB-for Engineers,Fourth Edition by Pearson.

Expected course outcomes: After the course, students will be

1. Able to use Matlab/Scilab for interactive computations.
2. Able to generate plots and export this for use in reports and presentations.
3. Able to program scripts and functions using the Matlab/Scilab development environment.
4. Able to use basic flow controls (if-else, for, while).
5. Familiar with strings and matrices and their use.

BTAM-18401

Discrete Structures (CSE & IT) 3L: 1T: 0P

Objectives

To provide knowledge of combinatorial problems, algebraic structures and graph theory required for building mathematical foundation of computer science.

Sets, Relations and Functions 6 hours

Basic operations and laws on sets, Cartesian products, Binary relation, Partial order relation, Equivalence Relation, different types of functions, their compositions and inverses.

Propositional Logic 6 hours

Syntax and semantics, proof systems, satisfiability, validity, soundness, completeness, deduction theorem, etc. Decision problems of propositional logic. Introduction to first order logic and first order theory.

Partially ordered sets 6 hours

Complete partial ordering, chain, lattice, complete, distributive, modular and complemented lattices, Boolean and pseudo Boolean lattices.

Introduction to Counting 6 hours

Basic counting techniques – inclusion and exclusion, pigeonhole principle, permutation, combination, Introduction to recurrence relation and generating functions.

Algebraic Structures 8 hours

Algebraic structures with one binary operation – semigroup, monoid and group. Cosets, Lagrange's theorem, normal subgroup, homomorphic subgroup. Congruence relation and quotient structures. Algebraic structures with two binary operations- ring, integral domain, and field. Boolean algebra and Boolean ring (Definitions and simple examples only).

Introduction to Graphs 6 hours

Graphs and their basic properties – degree, path, cycle, subgraph, isomorphism, Eulerian and Hamiltonian walk, trees.

Course Outcomes

Students will be able to:

- express logic sentence in terms of predicates, quantifiers, and logical connectives.
- derive the solution using deductive logic and prove the solution
- classify the algebraic structures of mathematical problems
- to evaluate Boolean functions and simplify expressions using the properties of Boolean Algebra
- to develop the given problem as graph networks and solve with techniques of graph theory.

Textbooks/References:

J. P. Tremblay and R. P. Manohar, Discrete Mathematics with Applications to Computer Science, Tata McGraw-Hill

C.L. Liu, Elements of Discrete Mathematics, Tata McGraw-Hill

R. C. Penner, Discrete Mathematics: Proof Techniques and Mathematical Structures, World Scientific

R. L. Graham, D. E. Knuth, and O. Patashnik, Concrete Mathematics, Addison-Wesley

K. H. Rosen, Discrete Mathematics and its Applications, Tata McGraw-Hill

J. L. Hein, Discrete Structures, Logic, and Computability, Jones and Bartlett

Narsingh Deo, Graph Theory with Application to Engineering and Computer Science, PHI

BTCS-18402
Computer Organization & Architecture

L:3 T:0 P:0

Objectives of the course:

To expose the students to the following:

1 Understanding of its various functional units of computer system.
2 Microprocessors, instruction execution and assembly level programming.
3 Fast Adders, Multiplication and Division Algorithms.
4 Control Unit Microprogramming, Hardwired control unit
5 Memory hierarchy, mapping and memory system design
6 IO Modes: Program control IO, DMA, Interupt initiated IO.
7 Basic concepts of pipelining, parallel processors and cache coherency.

Functional blocks of a computer: CPU, memory, input-output subsystems, control unit. Microprocessor based system design, Introduction to 8085 architecture, addressing modes, instruction set and instruction execution cycle.

Data Representation and Binary Arithmetic: Signed number representation, addition, subtraction, Booth multiplication algorithm, division algorithms, ripple adder, carry look ahead adders and array multipliers.

CPU control unit design: Hardwired and micro-programmed design approaches, Control Memory, RISC/CISC architecture.

Memory system design: Memory Hierarchy, memory organization, interleave memory, virtual memory, cache memory mapping techniques, and replacement algorithms, write policies.

IO Modes and Interfaces: Input-output subsystems, I/O device interface, I/O transfers – program controlled, interrupt driven and DMA, privileged and non-privileged instructions, software interrupts and exceptions. I/O device interface – ATA, SCSI, USB. Hard Disk Drive construction and working.

Pipelining: Basic concepts of pipelining, speedup efficiency and throughput, pipeline hazards.

Parallel Processors: Introduction to parallel processors, Concurrent access to memory and cache coherency.

Suggested books:

1. "Computer System Architecture", 3rd Edition by M. Morris Mano, Pearson Education India.
2. "Fundamentals of Microprocessor and Microcontrollers", by B Ram, Dhanpat Rai Publications.
3. "Computer Architecture and Organization", 3rd Edition by John P Hayes, McGraw Hill Education.
4. "Computer Organization and Architecture: Designing for Performance", 10th Edition by William Stallings, Pearson Education.

Course outcomes:

1. Draw the functional block diagram of bus architecture of a computer and describe the function of the instruction execution cycle, interpretation of instructions, addressing modes.
2. Implement assembly language program for given task like computing addition, subtraction, multiplication, division, searching, sorting etc.
3. Categorize memory organization and explain the function of each element of a memory hierarchy.
4. Identify and compare different methods for computer I/O mechanisms.
5. Write a flowchart for Concurrent access to memory and cache coherency in Parallel Processors and describe the process.

BTCS-18403

Operating Systems **L T P**
3 0 0

Objectives of the course

To learn the fundamentals of Operating Systems.

1. To learn the mechanisms of OS to handle processes and threads and their communication;
2. To learn the mechanisms involved in memory management in contemporary OS;
3. To gain knowledge on distributed operating system concepts that includes architecture, Mutual exclusion algorithms, deadlock detection algorithms and agreement protocols ; and
4. To know the components and management aspects of concurrency management.

Module 1:

Introduction: Concept of Operating Systems, Generations of Operating systems, Types of Operating Systems, OS Services, System Calls, Structure of an OS - Layered, Monolithic, Microkernel Operating Systems, Concept of Virtual Machine. Case study on UNIX and WINDOWS Operating System.

Module 2:

Processes: Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching
Thread: Definition, Various states, Benefits of threads, Types of threads, Concept of multithreads,
Process Scheduling: Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time; Scheduling algorithms: Pre-emptive and Non pre-emptive, FCFS, SJF, RR; Multiprocessor scheduling: Real Time scheduling: RM and EDF.
Module 3:
Inter-process Communication: Critical Section, Race Conditions, Mutual Exclusion, Hardware Solution, Strict Alternation, Peterson's Solution, The Producer\ Consumer Problem, Semaphores, Event Counters, Monitors, Message Passing, Classical IPC Problems: Reader's & Writer Problem, Dinning Philosopher Problem etc.

Module 4:
Deadlocks: Definition, Necessary and sufficient conditions for Deadlock, Deadlock Prevention, Deadlock Avoidance: Banker's algorithm, Deadlock detection and Recovery.

Module 5:
Memory Management: Basic concept, Logical and Physical address map, Memory allocation: Contiguous Memory allocation – Fixed and variable partition–Internal and External fragmentation and Compaction; Paging: Principle of operation – Page allocation – Hardware support for paging, Protection and sharing, Disadvantages of paging.
Virtual Memory: Basics of Virtual Memory – Hardware and control structures – Locality of reference, Page fault , Working Set , Dirty page/Dirty bit – Demand paging, Page Replacement algorithms: Optimal, First in First Out (FIFO), Second Chance (SC), Not recently used (NRU) and Least Recently used (LRU).

Module 6:
I/O Hardware: I/O devices, Device controllers, Direct memory access Principles of I/O Software: Goals of Interrupt handlers, Device drivers, Device independent I/O software, Secondary-Storage Structure: Disk structure, Disk scheduling algorithms
File Management: Concept of File, Access methods, File types, File operation, Directory structure, File System structure, Allocation methods (contiguous, linked, indexed), Free-space management (bit vector, linked list, grouping), directory implementation (linear list, hash table), efficiency and performance.

Module 7:
Disk Management: Disk structure, Disk scheduling - FCFS, SSTF, SCAN, C-SCAN, Disk reliability, Disk formatting, Boot-block, Bad blocks

Suggested books:

1. Operating System Concepts Essentials, 9th Edition by AviSilberschatz, Peter Galvin, Greg Gagne, Wiley Asia Student Edition.
2. Operating Systems: Internals and Design Principles, 5th Edition, William Stallings, Prentice Hall of India.

Suggested reference books:

1. Operating System: A Design-oriented Approach, 1st Edition by Charles Crowley, Irwin Publishing
2. Operating Systems: A Modern Perspective, 2nd Edition by Gary J. Nutt, Addison-Wesley
3. Design of the Unix Operating Systems, 8th Edition by Maurice Bach, Prentice-Hall of India
4. Understanding the Linux Kernel, 3rd Edition Daniel P. Bovet, Marco Cesati, O'Reilly and Associates

Course Outcomes:

1. Create processes and threads;
2. Develop algorithms for process scheduling for a given specification of CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time.
3. For a given specification of memory organization develop the techniques for optimally allocating memory to processes by increasing memory utilization and for improving the access time.
4. Design and implement file management system.
5. For a given I/O devices and OS (specify) develop the I/O management functions in OS as part of a uniform device abstraction by performing operations for synchronization between CPU and I/O controllers.

BTCS-18404

Design and Analysis of algorithm

L T P
3 0 0

Objectives of the course

1. Analyze the asymptotic performance of algorithms.
2. Write rigorous correctness proofs for algorithms.
3. Demonstrate a familiarity with major algorithms and data structures.
4. Apply important algorithmic design paradigms and methods of analysis.
5. Synthesize efficient algorithms in common engineering design situations.

Module 1:
Introduction: Characteristics of algorithm. Analysis of algorithm: Asymptotic analysis of complexity bounds – best, average and worst-case behavior; Performance measurements of Algorithm, Time and space trade-offs, Analysis of recursive algorithms through recurrence relations: Substitution method, Recursion tree method and Masters' theorem.

Module 2:
Fundamental Algorithmic Strategies: Brute-Force, Greedy,Dynamic Programming, Branch-and-Bound and Backtracking methodologies for the design of algorithms; Illustrations of these techniques for Problem-Solving , Bin Packing, Knap Sack TSP. Heuristics – characteristics and their application domains.

Module 3:
Graph and Tree Algorithms: Traversal algorithms: Depth First Search (DFS) and Breadth First Search (BFS); Shortest path algorithms, Transitive closure, Minimum Spanning Tree, Topological sorting, Network Flow Algorithm.

Module 4:
Tractable and Intractable Problems: Computability of Algorithms, Computability classes – P, NP, NP-complete and NP-hard. Cook's theorem, Standard NP-complete problems and Reduction techniques.

Module 5:
Advanced Topics: Approximation algorithms, Randomized algorithms, Class of problems beyond NP – P SPACE

Suggested books:
1 Introduction to Algorithms, 4TH Edition, Thomas H Cormen, Charles E Lieserson, Ronald L Rivest and Clifford Stein, MIT Press/McGraw-Hill.
2 Fundamentals of Algorithms – E. Horowitz et al.

Suggested reference books
1 Algorithm Design, 1ST Edition, Jon Kleinberg and ÉvaTardos, Pearson.
2 Algorithm Design: Foundations, Analysis, and Internet Examples, Second Edition, Michael T Goodrich and Roberto Tamassia, Wiley.

Course Outcomes

1. For a given algorithms analyze worst-case running times of algorithms based on asymptotic analysis and justify the correctness of algorithms .
2. Describe the greedy paradigm and explain when an algorithmic design situation calls for it. For a given problem develop the greedy algorithms.
3. Describe the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it. Synthesize divide-and-conquer algorithms. Derive and solve recurrence relation.
4. Describe the dynamic-programming paradigm and explain when an algorithmic design situation calls for it. For a given problems of dynamic-programming and develop the dynamic programming algorithms, and analyze it to determine its computational complexity.
5. For a given model engineering problem model it using graph and write the corresponding algorithm to solve the problems.
6. Explain the ways to analyze randomized algorithms (expected running time, probability of error).

BTHS-18904

Organisational Behaviour

Credit:3

L T P

3 0 0

Internal Marks: 40

External Marks: 60

Total Marks: 100

UNIT 1:

Introduction to Organisational Behaviour: the OB Model, Roles of Manager in OB, Douglas McGregor Theory X and Theory Y.

UNIT2:

Foundation of Individual Behaviour: Concepts of Motivation, Personality, Values, Attitudes, Perception, Learning, Individual Decision-Making and Problem-Solving.

UNIT3:

Foundation of Group Behaviour: Concepts related to Communication, Concept of leadership and styles, Work Teams and Group Dynamics.

UNIT4:

Foundation of the Organisation: Concepts related to Organisation Structure, Organization Culture, Organizational Conflict and Discipline.

UNIT 5:

Organisation Management: Definition of management, Function of Management, Maslow Hierarchy, Principles of Henry Fayol and F.W Taylor.

Suggested Reading:

Organizational Behaviour – Stephen P. Robbins, Timothy A.Judge, SeemaSanghi

Organisational Behaviour – L M Prasad

Organizational Behavior, Human Behavior At Work – John W Newstorm

Management &Organisational Behaviour – Laurie J Mulli

BTCS-18405
Computer Organization & Architecture Lab

L:2 T:0 P:2

Objective: To understand the basic concepts of Computer Organization and Assembly Language Programming.

1. Familiarization with Computer Hardware and Peripherals.
2. Familiarization with 8085 Microprocessor kit.
3. Write an ALP to add of two 8 bit numbers to get 8 bit sum.
4. Write a ALP to add two 8 bit numbers to get 16 bit sum.
5. Write an ALP to add two 16-bit numbers to get 16-bit sum.
6. Write an ALP to add two 8-bit BCD Numbers.
7. Write an ALP to exchange the contents of BC pair with DE Pair.
8. Write an ALP to find 1's complement and 2's complement of an 8 bit number.
9. Write an ALP to subtract two signed 8-bit numbers.
10. Write an ALP to find sum of "n" 8 bit numbers.
11. Write an ALP to multiply an 8 bit number by 2 using shifting method.
12. Write an ALP to find larger of two 8 bit numbers.
13. Write an ALP to find largest number among "n" 8-bit numbers.
14. Write an ALP to sort "n" numbers in ascending order.

Course Outcome (CO):

1. To get familiar with various parts of computer system and peripherals..
2. To get familiar with Microprocessor, PIN structure, use of opcodes, kit details etc.
3. To know the use of various instructions, stack pointer and flags etc.
4. Write ALPs to do logical, arithmetic, searching and sorting operations.

BTCS-18406
Operating System Lab

Objective: This course provides knowledge of different operating systems.

1. Installation and Configuration of latest Windows Operating System;
2. Installation and Configuration of latest UNIX Operating System (any flavor)
3. Practice of using various editors for example vim, ex, and ed;
4. Practice of commands used in networking for example ifconfig, ipconfig, traceroute, telnet, nslookup, netstat, scp, nmap, ping;
5. Practice of directory listing commands with various options for example ls, dir
6. Practice of process related commands for example, ps, who, kill, sleep;
7. Practice of backup and recovery commands for example tar, cpio;
8. Practice of file management commands for example cd, cp, rm, mkdir, rmdir, cat, ws, sat, cut, grep, chmod,, chown, dd, df;
9. Printing commands, grep, fgrep, find, sort, cal, banner, touch file,
10. Shell Programming: Basic of shell programming, various types of shell, Shell Programming in bash, conditional & looping statement, case statements, parameter passing and arguments, shell variables, shell keywords, creating shell programs for automate system tasks, report printing.
11. Virtualization, Installation of Virtual Machine Software such as vmware / virtualbox and installation of various operating System on Virtual Machine;

COURSE OUTCOMES (CO): The expected outcomes are:

1. Installation and configuration of Windows and UNIX Operating Systems.
2. Practice of various commands in Windows and Unix Operating Systems
3. Learning of Shell Programming; and
4. Knowledge of virtualization.

BTCS-18407
Design & Analysis of Algorithms Lab

Objective: To get a first-hand experience of implementing well-known algorithms in a high-level language. And to be able to compare the practical performance of different algorithms for the same problem.

1. Code and analyze to compute the greatest common divisor (GCD) of two numbers.

2. Code and analyze to find the median element in an array of integers.

3. Code and analyze to find the majority element in an array of integers.

4. Code and analyze to sort an array of integers using Heap sort.

5. Code and analyze to sort an array of integers using Merge sort.

6. Code and analyze to sort an array of integers using Quick sort.

7. Code and analyze to find the edit distance between two character strings using dynamic programming. 8. Code and analyze to find an optimal solution to weighted interval scheduling using dynamic programming.

9. Code and analyze to find an optimal solution to matrix chain multiplication using dynamic programming.

10. Code and analyze to do a depth-first search (DFS) on an undirected graph. Implementing an application of DFS such as (i) to find the topological sort of a directed acyclic graph, OR (ii) to find a path from source to goal in a maze.

11. Code and analyze to do a breadth-first search (BFS) on an undirected graph. Implementing an application of BFS such as (i) to find connected components of an undirected graph, OR (ii) to check whether a given graph is bipartite.

12. Code and analyze to find the minimum spanning tree in a weighted, undirected graph.

13. Code and analyze to find all occurrences of a pattern P in a given string S.

14. Code and analyze to multiply two large integers using Karatsuba algorithm.

15. Code and analyze to compute the convex hull of a set of points in the plane.

16. (Mini-project Topic) Program to multiply two polynomials using Fast Fourier Transform.

COURSE OUTCOMES (CO): The students are expected to:

1. Write code for different methods of computation;

2. Practice of writing program different searching and sorting algorithms and do analysis of the same; and

3. Coding in dynamic programming.

BTHS-18905
Effective Technical Communication

L:3, T:,P:0

Internal Marks: 40
External Marks: 60
Total Marks: 100

Module 1: Information Design and Development- Different kinds of technical documents, Information development life cycle, Organization structures, factors affecting information and document design, Strategies for organization, Information design and writing for print and for online media.

(7hrs)

Module 2: Technical Writing, Grammar and Editing- Technical writing process, forms of discourse, Writing drafts and revising, Collaborative writing, creating indexes, technical writing style and language. Basics of grammar, study of advanced grammar, editing strategies to achieve appropriate technical style. Introduction to advanced technical communication, Usability, Hunan factors, Managing technical communication projects, time estimation, Single sourcing, Localization.

(7hrs)

Module 3: Self Development and Assessment- Self assessment, Awareness, Perception and Attitudes, Values and belief, Personal goal setting, career planning, Self-esteem. Managing Time; Personal memory, Rapid reading, Taking notes; Complex problem solving; Creativity.

(7hrs)

Module 4: Communication and Technical Writing- Public speaking, Group discussion, Oral; presentation, Interviews, Graphic presentation, Presentation aids, Personality Development. Writing reports, project proposals, brochures, newsletters, technical articles, manuals, official notes, business letters, memos, progress reports, minutes of meetings, event report.

(7hrs)

Module 5: Ethics- Business ethics, Etiquettes in social and office settings, Email etiquettes, Telephone Etiquettes, Engineering ethics, Managing time, Role and responsibility of engineer, Work culture in jobs, Personal memory, Rapid reading, Taking notes, Complex problem solving, Creativity.

(7hrs)

Books Recommended:-

1. David F. Beer and David McMurrey, Guide to writing as an Engineer, John Willey. New York, 2004
2. Shiv Khera, You Can Win, Macmillan Books, New York, 2003.
3. Raman Sharma, Technical Communications, Oxford Publication, London, 2004.
4. Sharma, R. and Mohan, K. Business Correspondence and Report Writing, TMH New Delhi 2002.

BTCS-H-18101

Graph Theory

	L	T	P
Internal Marks: 40	**3**	**0**	**0**
External Marks: 60			
Total Marks: 100			

UNIT I

Introduction – Graph Terminologies – Definitions of Graph, Types of Graphs – Sub Graph- Multi Graph – Regular Graph – Isomorphism – Isomorphic Graphs – Sub-graph – Euler graph – Hamiltonian Graph

UNIT II

Trees -Properties- Distance and Centres – Types – Rooted Tree– Tree Enumeration- Labeled Tree – Unlabeled Tree – Spanning Tree – Fundamental Circuits- Cut Sets – Properties – Fundamental Circuit and Cut-set- Connectivity- Separability -Related Theorems.

UNIT III

Network Flows – Planar Graph – Representation – Detection – Dual Graph – Geometric and Combinatorial Dual – Related Theorems – Digraph – Properties – Euler Digraph.

UNIT IV

Matrix Representation – Adjacency matrix- Incidence matrix- Circuit matrix – Cut-set matrix – Path Matrix- Properties – Related Theorems – Correlations. Graph Coloring

UNIT V

Graph Algorithms- Connectedness and Components- Spanning Tree- Fundamental Circuits- Cut Vertices- Directed Circuits- Shortest Path – Applications overview.

Course OUTCOMES:

Upon completion of this course, the students should be able to

1. Understand the basic concepts of graphs, and different types of graphs
2. Understand the properties, theorems and be able to prove theorems.
3. Apply suitable graph model and algorithm for solving applications.

TEXT BOOKS:

Narsingh Deo, “Graph Theory with Application to Engineering and Computer Science”, Prentice-Hall of India Pvt.Ltd, 2003.
L.R.Foulds , “Graph Theory Applications”, Springer ,2016.

REFERENCES:

Bondy, J. A. and Murty, U.S.R., “Graph Theory with Applications”, North Holland Publication,2008.
West, D. B., ―Introduction to Graph Theory, Pearson Education, 2011.
John Clark, Derek Allan Holton, ―A First Look at Graph Theory‖, World Scientific Publishing Company, 1991.
Diestel, R, “Graph Theory”, Springer,3rd Edition,2006.
Kenneth H.Rosen, “Discrete Mathematics and Its Applications”, Mc Graw Hill ,

BTCS-H-18104 Software Project Management

L T P
3 0 0

Internal Marks: 40
External Marks: 60
Total Marks: 100

Objective- Software development is a complex process involving such activities as domain analysis, requirements specification, communication with the customers and end-users, designing and producing different artifacts, adopting new paradigms and technologies, evaluating and testing software products etc. The proposed subject will take students through the various processes involved in project management.

Contents:

Project Evaluation and Planning - Activities in Software Project Management, Overview of Project Planning, Stepwise planning, contract management, Software processes and process models. Cost Benefit Analysis, Cash Flow Forecasting,

Cost-Benefit Evaluation Techniques, Risk Evaluation. Project costing, COCOMO 2, Staffing pattern, Effect of schedule compression, Putnam equation, Capers Jones estimating rules of thumb, Project Sequencing and Scheduling Activities, Scheduling resources, Critical path analysis, Network Planning, Risk Management, Nature and Types of Risks, Managing Risks, Hazard Identification, Hazard Analysis, Risk Planning and Control.

Monitoring And Control- Collecting Data, Visualizing Progress, Cost Monitoring, review techniques, project termination review, Earned Value analysis, Change Control, Software Configuration Management (SCM).

Quality Management and People Management- Introduction, Understanding Behavior, Organizational Behaviour, Selecting The Right Person For The Job, Motivation, The Oldman – Hackman Job Characteristics Model , Working in Groups, Organization and team structures, Decision Making, Leadership.

Suggested Readings/Books

1. Bob Hughes, Mike Cotterell, “Software Project Management”, Tata McGraw Hill.

2. Royce, “Software Project Management”, Pearson Education.

3. Robert K. Wysocki, “Effective Software Project Management”, Wiley

4. Ian Sommerville, Software Engineering, Seventh Edition, Pearson Education.

5. R.S. Pressman, Software Engineering: A Practitioner's Approach, Sixth Edition, Tata McGraw-Hill.

COURSE OUTCOMES (CO): The students are expected to:-

1.Understand the concepts of developing a software project

2.Understand the cost benefit analysis of software projects

3.Know the monitoring and control methods during progress of a project

BTCS-H-18105 CRYPTOGRAPHY AND NETWORK SECURITY

L	T	P
3	0	0

Objectives:

Upon completion of this course, students will have gained knowledge of

1. Cryptographic concepts;
2. Network Security principles and approaches; and
3. System Security Concepts.

Symmetric Ciphers - Overview: Services, Mechanisms and Attacks, The OSI Security Architecture, A Model of Network Security. Classicial Encryption Techniques: Symmetric Cipher Model, Substitution Techniques, Transposition Techniques, Rotor Machines, Steganography. Block Cipher and the Data Encryption Standard: Simplified DES, Block Cipher Principles, The DES, The Strength of DES, Differential and Linear Cryptanalysis. Symmetric Ciphers: Triple DES, Blowfish. Confidentiality using Conventional Encryption: Placement of Encryption Function, Traffic Confidentiality, Key Distribution, Random Number Generation. [10]

Public Key Encryption, Digital Signatures - Number Theory, Prime Numbers Format‟s and Euler‟s Theorems, Testing for Primality. Public Key Cryptography and RSA: Principles of Public Key Cryptosystems, The RSA Algorithms, Key Management, Diffie Hellman Key Exchange. [8]

Authentication Protocols - Message Authentication: Authentication Requirements, Authentication Functions, Message Authentication Codes, MD5 Message Digest Algorithms, Digital Signatures and Authentication Protocols: Digital Signatures, Authentication Protocols, Digital Signature Standards. [7]

Network Security - Authentication Applications: Kerberos, X.509 Directory Authentication Service. Electronic Mail Security: Pretty Good Privacy. IP Security: Overview, IP Security Architecture, Authentication Header, Encapsulation Security Payload. Web Security: Web Security Requirements, Secure Sockets Layer and Transport Layer Security, Secure Electronic Transaction. [5]

System Security- Intruders, Malicious Software, Viruses and Related Threats, Counter Measures, Firewalls and its Design Principles. [5]

Text Books:

1. Cryptography and Network Security by Behrouz A. Forouzan, Tata McGraw Hill.
2. William Stallings, Cryptography and Network Security Principles and practice. 2/e,Pearson Education.
3. William Stallings, Network Security Essentials, Applications and Standards Pearson Education.
4. Atul Kahate Cryptography & Network Security, TMH, 2nd Edition

Reference Books:

1. Charlie Kaufman, Radia Perlman, Mike Speciner, Network Security: Private Communication in Public World, 2nd Edition, 2011, Pearson Education.
2. Bruce Schneier, “Applied Cryptography” , John Wiley and Sons, Inc.
3. Bishop, Matt, Introduction to Computer Security. Addison-Wesley, Pearson Education, Inc. ISBN: 0-321-24744-2. (2005)
4. Michael. E. Whitman and Herbert J. Mattord Principles of Information Security, Cengage Learning

COURSE OUTCOMES (CO): The students are expected to:

1. Understand the concepts of cryptography;
2. Get the knowledge of IP Security, Web Security, and E-mail Security;
3. Know the working of SSL and SET protocols; and
4. Understand the system Security concepts.

BTCS-H-18106 (Internet-of-Things)

L T P
3 0 0

Internal Marks: 40
External Marks: 60
Total Marks: 100

Objective of the course: The basic objective of this course in to introduce the basic concepts of IOT to undergraduate students.

Detailed contents:

Module1:
Introduction: Internet of things, IOT conceptual framework, IOT architectural view, Technology behind IOT, Sources of IOT, some examples of IOT

Module2:

Design Principles: IOT/M2M system layers and design standardization, Communication technologies, Data enrichment, data consolidation and device management at gateway.

Module3:
Internet connectivity principles: Internet connectivity, internet based communication, IP addressing in IOT

Module4:
Data Acquisition and storage, organizing data, Analysis of data, Different phases of analysis, predictive analysis, Prescriptive analysis

Module 5:
Sensor technology, Participatory Sensing, Actuator, sensor data communication protocols, RFID technology

Suggested books:

1."Internet of Things, Architecture and design principles", by Raj Kamal, Mc Graw Hill.
2. "internet of Things: A Hands-On Approach" by Arshdeep Bahga, Universities Press

Course outcomes : After the completion of this course, the students will be able to:

1 To understand the basic concepts of Internet of Things (IOT).

2 To understand concepts about designing the IOT networks.

3 To understand basic concepts about Data acquiring and analysis in IOT applications..

4 To understand basic concepts about sensor technologies.