DAV INSTITUTE OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
Scheme and Syllabus

Third Semester — **Contact Hours: 30 Hrs.**

Course Code	Course Name	Load Allocation			Marks Distribution		Total Marks	Credits
		L	T	P	Internal	External		
BTCS301	Computer Architecture	3	1	-	40	60	100	4
BTAM302	Mathematics –III	3	1	-	40	60	100	4
BTCS303	Digital Circuits & Logic Design	3	1	-	40	60	100	4
BTCS304	Data Structures	3	1	-	40	60	100	4
BTCS305	Object Oriented Programming using C++	3	1	-	40	60	100	4
BTCS306	Data Structures Lab	-	-	4	30	20	50	2
BTCS307	Institutional Practical Training#	-	-	-	60	40	100	1
BTCS308	Digital Circuits & Logic Design Lab	-	-	2	30	20	50	1
BTCS309	Object Oriented Programming using C++ Lab	-	-	4	30	20	50	2
	Total	15	5	10	350	400	750	26

The marks will be awarded on the basis of 04 weeks Institutional Practical training conducted after 2nd semester

Fourth Semester — **Contact Hours: 30 Hrs.**

Course Code	Course Name	Load Allocation			Marks Distribution		Total Marks	Credits
		L	T	P	Internal	External		
BTCS401	Operating Systems	3	1	-	40	60	100	4
BTCS402	Discrete Structures	3	1	-	40	60	100	4
BTCS403	Computer Networks-I	3	1	-	40	60	100	4
BTCS404	Microprocessor& Assembly Language Programming	3	1	-	40	60	100	4
BTCS405	System Programming	3	1	-	40	60	100	4
BTCS406	Operating System Lab	-	-	2	30	20	50	1
BTCS407	Computer Networks-I Lab	-	-	4	30	20	50	2
BTCS408	Microprocessor& Assembly Language Programming Lab	-	-	2	30	20	50	1
BTCS409	System Programming Lab	-	-	2	30	20	50	1
General Fitness					100	-	100	-
	Total	15	5	10	420	380	800	25

Fifth Semester — **Contact Hours: 29 Hrs.**

Course Code	Course Name	Load Allocation			Marks Distribution		Total Marks	Credits
		L	T	P	Internal	External		
BTCS501	Computer Networks –II	3	1	-	40	60	100	4
BTCS502	Relational Database Management System	3	1	-	40	60	100	4
BTCS503	Design & Analysis of Algorithms	3	1	-	40	60	100	4
BTCS504	Computer Graphics	3	1	-	40	60	100	4
BTCS505	Computer Peripherals & Interfaces	3	0	-	40	60	100	3
BTCS506	RDBMS Lab	-	-	4	30	20	50	2
BTCS507	Computer Networks –II Lab	-	-	2	30	20	50	1
BTCS508	Design & Analysis of Algorithms Lab	-	-	2	30	20	50	1
BTCS509	Computer Graphics Lab	-	-	2	30	20	50	1
BTCS510	Industrial Training*	-	-	-	60	40	100	1
	Total	15	4	10	380	420	800	25

*The marks will be awarded on the basis of 06 weeks industrial training conducted after 4th semester

Sixth Semester **Contact Hours: 30 Hrs.**

Course Code	Course Name	Load Allocation			Marks Distribution		Total Marks	Credits
		L	T	P	Internal	External		
BTCS601	Simulation and Modeling	3	-	-	40	60	100	3
BTCS602	RDBMS -II	3	1	-	40	60	100	4
BTCS603	Software Engineering	3	-	-	40	60	100	3
BTCSXXX	Elective –I	3	1	-	40	60	100	4
BT***	Open Elective	3	1	-	40	60	100	4
BTCS604	RDBMS-II Lab	-	-	4	30	20	50	2
BTCS605	Free/ Open Source Software Lab	-	-	4	30	20	50	2
BTCS606	Software Engineering Lab	-	-	2	30	20	50	1
BTCS607	Simulation and Modeling Lab	-	-	2	30	20	50	1
General Fitness					100	-	100	
Total		15	3	12	420	380	800	24

Seventh Semester / Eighth Semester **Contact Hours: 29 Hrs**

Course Code	Course Name	Load Allocation			Marks Distribution		Total Marks	Credits
		L	T	P	Internal	External		
BTCS701	Artificial Intelligence	3	-	-	40	60	100	3
BTCS702	Theory of Computation	3	1	-	40	60	100	4
BTCS 703	Project	-	-	12	150	150	300	12
BTCSYYY	Elective –II	3	1	-	40	60	100	4
BTCSZZZ	Elective –III	3	1	-	40	60	100	4
BTCS704	Artificial Intelligence Lab	-	-	2	30	20	50	1
General Fitness					100	-	100	
Total		12	03	14	440	410	850	28

Course Code	Course Name	Marks Distribution		Total Marks	Credits
		Internal	External		
BTCS801	Software Training	150	100	250	8
BTCS802	Industry Oriented Project Training	300	200	500	10
Total		450	300	750	18

LIST OF ELECTIVES:

Elective –I BTXS XXX

- BTCS 901 Web Technologies
- BTCS 902 Mobile Applications Development
- BTCS 903 Ethical Hacking
- BTCS 904 Information Security

Elective –II BTCS YYY

- BTCS 905 Software Testing and Quality Assurance
- BTCS 906 Object Oriented Analysis and Design
- BTCS 907 Software Project Management
- BTCS 908 Business Intelligence
- BTCS 909 Agile Software Development

Elective -III BTCS ZZZ

- BTCS 910 Multimedia and Application
- BTCS 911 Soft Computing
- BTCS 912 Cloud Computing
- BTCS 913 Compiler Design
- BTCS 914 Big Data
- BTCS 915 Digital Image Processing
- BTCS 916 Enterprise Resource Planning

Third Semester

COMPUTER ARCHITECTURE
BTCS-301

COURSE OBJECTIVES

This course offers a good understanding of the various functional units of a computer system and familiarization of the students with design and organization of digital computers.

CONTENTS

1. **Register Transfer and Microoperations**: Register transfer language & operations, arithmetic microoperations, logic microoperations, shift microoperations, arithmetic logic shift unit. Design of a complete basic computer and its working.
2. **Basic Computer Organisation and Design**: Instruction codes, Computer registers, Computer Instructions, Timing and control, Instruction Cycle, Memory reference instructions, Input/ Output and Interrupt, Design of basic Computer, Design of Accumulator Logic.
3. **Design of Control Unit:** Control memory, design of control unit – microprogrammed, hardwired, and their comparative study.
4. **Central Processing Unit**: General Register Organisation, Stack Organisation, Instruction formats, Addressing Modes, Data transfer and manipulations, Program control, RISC and CISC architecture.
5. **Input-Output Organisation:** Peripheral devices, I/O Interface, asynchronous data transfer, modes of transfer, priority interrupt, DMA, I/O processor, serial communication.
6. **Memory Organisation**: Memory hierarchy, main memory, auxiliary memory, associative memory, cache memory, virtual memory, memory management hardware.
7. **Advanced concepts of Computer Architecture:** Concept of pipeline, Arithmetic pipeline, Instruction, vector processors and array processors. Introduction to parallel processing, Interprocessor communication & synchronization.

COURSE OUTCOMES

After the completion of the course the student will be able:

CO1 To Understand the basic architectural design and working of functional units of a computer system.

CO2 To Understand Memory and Input-Output organization of a digital computer and analyze related performance issues.

CO3 To Demonstrate knowledge of methods for processor performance improvement such as pipelining, vector and array processing.

Suggested Readings/ Books:

1. M. Moris Mano, Computer System Architecture, Pearson Education.
2. William Stallings, Computer Organisation and Architecture, Pearson Education.
3. David A Patterson, Computer Architecture, Pearson Education.
4. P. Pal Choudhri, Computer Organisation and Design, PHI.
5. J. P. Hayes, Computer System Architecture, Pearson Education.
6. Kai Hawang, Advanced Computer Architecture, Tata McGraw Hill.
7. Riess. Assembly Language and Computer Architecture and using C++ and JAVA, Cengage Learning.

ENGINEERING MATHEMATICS-III

BTAM-302

COURSE OBJECTIVES: To teach computer based Engineering Mathematics to students. After this course the student will be able to solve computer oriented problems.

CONTENTS:

Fourier series: Periodic Functions, Euler's Formula. Even and odd Functions, Half range expansions, Fourier series of different waveforms.

Laplace transformations: Laplace transforms of various standard functions, properties of Laplace transform.

Partial Differential Equations: Formation of Partial Differential Equations, linear Partial Differential Equations, Homogeneous Partial Differential Equations with constant coefficients.

Functions of complex variables: Limits, continuity and derivatives of the function of complex variables, Analytic function, Cauchy- Riemann equations, conjugate functions.

Linear Systems and Eigen- Values: Gauss – elimination method, gauss- Jordan method, Gauss-Seidel iteration method, Rayleigh's Power method for Eigen values and Eigenvectors.

Differential Equations: Solutions of Initial values problems using Eulers, modified Eulers method and Runge- kutta (upto fourth order) methods.

Probability distribution: Binomial, Poisson and Normal distribution.

Sampling Distribution & testing of Hypothesis: Sampling, Distribution of means and variance, Chi-Square distribution, t- distribution, F- distribution. General concepts of hypothesis, Testing a statistical Hypothesis, One and two tailed tests, critical region, Confidence interval estimation. Single and two sample tests on proportion, mean and variance.

COURSE OUTCOMES

CO1. Demonstrate their understanding of the Dirichlet's conditions and evaluation of Fourier series for standard periodic waveforms.

CO2. Calculate the Laplace transform of standard functions both from the definition and by using laws.

CO3. Check the condition for a complex variable function to be analytic and/or harmonic & find complex conjugates.

CO4. Demonstrate finding solutions of linear systems & Eigen values and differential equations using Numerical methods.

CO5. Apply the concept of probability and statistics to find probability distributions and apply them for hypothesis testing.

TEXT/ REFERENCE BOOKS:

1. Engineering Mathematics by Manish Goyal, Satyaparkashan Publication, New Delhi
2. Advanced Engineering Mathematics by Kreyszing Erwin ; Wiley Eastern, New Delhi
3. Higher Engineering Mathematics by BS Grewal : Khanna Publishers, New Delhi.
4. Differential Equations by Sharma and Gupta ; Krishna Prakashan Media (P) Ltd., Meerut.
5. Engineering Mathematics by N. P. Bali, Usha Paul ; Laxmi Publcation, New Delhi
6. Peter V.O'Neil," Advanced Engineering Mathematics", Cengage Learning
7. P. E. Danko, A. G. Popov, T. Y. A. Kaznevnikova, " Higher Mathematics in Problems and Exercise", Part 2, Mir Publishers, 1983.

DIGITAL CIRCUITS & LOGIC DESIGN
BTCS-303

COURSE OBJECTIVES :

1. This course is designed to provide a comprehensive introduction to digital circuit and logic design leading to the ability to understand number system representations, binary codes, binary arithmetic and Boolean algebra, its axioms and theorems and its relevance to digital logic design.
2. Introduction to combinational circuits (such as K-Maps), and sequential circuits (such as flip-flops).
3. Analyse and design simple systems composed of programmable logic such as ROMs and PLAs.

CONTENTS:

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

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

3. 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.

4. 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.

5. 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. Classification of sequential circuits-Moore and Mealy, Design of Synchronous counters: state diagram, Circuit implementation. Shift registers

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

7. Signal Conversions: Analog & Digital signals. A/D and D/A conversion techniques (Weightedtype, R-2R Ladder type, Counter Type, Dual Slope type, Successive Approximation type).

COURSE OUTCOMES

Upon completion of the subject, Students will be able to

CO1. Demonstrate knowledge of binary number theory, binary codes and Boolean algebra.

CO2. Perform basic arithmetic operations with signed integers represented in binary.

CO3. Analyze and design combinational systems using standard gates and minimization methods (such as K-Maps).

CO4. Analyze and design combinational systems composed of standard combinational modules such as multiplexers and decoders.

CO5. Demonstrate knowledge of sequential circuits and design flip-flops, counters and shift registers.

CO6. Analyze and design simple systems composed of programmable logic such as ROMs or PLAs.

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, 5 ed., Tata McGraw Hill Publishing Company Limited, New Delhi, 2003.
3. R.P.Jain, Modern Digital Electronics, 3 ed., Tata McGraw–Hill publishing company limited, New Delhi, 2003.
4. Thomas L. Floyd, Digital Fundamentals, Pearson Education, Inc, New Delhi, 2003
5. Ronald J. Tocci, Neal S. Widmer, Gregory L. Moss, Digital System -Principles and Applications, Pearson Education.
6. Ghosal , Digital Electronics, Cengage Learning.

DATA STRUCTURES

BTCS-304

COURSE OBJECTIVES

The graduate will be able:

1. To understand the basic fundamentals of different types of data structures.
2. To design and implementation of various linear and non linear data structures.
3. To understand and calculate the time complexity of algorithms.
4. To understand basic searching and sorting algorithms.

CONTENTS:

Dynamic Memory Management: Understanding pointers, usage of pointers, arithmetic on pointers, memory allocation, memory management functions and operators, debugging pointer - dangling pointers, memory leaks, etc.
Introduction: Concept of data type, definition and brief description of various data structures, data structures versus data types, operations on data structures, algorithm complexity, Big O notation.
Arrays: Linear and multi-dimensional arrays and their representation, operations on arrays, sparse matrices and their storage.
Linked List: Linear linked list, operations on linear linked list, doubly linked list, operations on doubly linked list, application of linked lists.
Stacks: Sequential and linked representations, operations on stacks, application of stacks such as parenthesis checker, evaluation of postfix expressions, conversion from infix to postfix representation, implementing recursive functions.
Queues: Sequential representation of queue, linear queue, circular queue, operations on linear and circular queue, linked representation of a queue and operations on it, deque, priority queue, applications of queues.

Trees: Basic terminology, sequential and linked representations of trees, traversing a binary tree using recursive and non-recursive procedures, inserting a node, deleting a node, brief introduction to threaded binary trees, AVL trees and B-trees.
Heaps: Representing a heap in memory, operations on heaps, application of heap in implementing priority queue and heap sort algorithm.
Graphs: Basic terminology, representation of graphs (adjacency matrix, adjacency list), traversal of a graph (breadth-first search and depth-first search), and applications of graphs.
Hashing & Hash Tables: Comparing direct address tables with hash tables, hash functions, concept of collision and its resolution using open addressing and separate chaining, double hashing, rehashing.
Searching & Sorting: Searching an element using linear search and binary search techniques, sorting arrays using bubble sort, selection sort, insertion sort, quick sort, merge sort, heap sort, shell sort and radix sort, complexities of searching & sorting algorithms.

COURSE OUTCOMES

After competition of course the student will able to:

1. Gain extensive knowledge on basic principles and algorithms of data structures
2. Understand arrays, linked-list, stack and queues, trees, graphs
3. Compare performance of different sorting algorithms.

Suggested Readings/ Books:

1. Sartaj Sahni, Data Structures, Algorithms and Applications in C++, Tata McGraw Hill.
2. Tenenbaum, Augenstein, & Langsam, Data Structures using C and C++, Prentice Hall of India.
3. R. S. Salaria, Data Structures & Algorithms Using C++, Khanna Book Publishing Co. (P) Ltd.
4. Seymour Lipschutz, Data Structures, Schaum's Outline Series, Tata McGraw Hill
5. Kruse, Data Structures & Program Design, Prentice Hall of India.
6. Michael T. Goodrich, Roberto Tamassia, & David Mount, Data Structures and Algorithms in C++ (Wiley India)

OBJECT ORIENTED PROGRAMMING USING C++
BTCS-305

COURSE OBJECTIVES:

1) To understand the difference between structured programming and object oriented programming paradigm.
2) To understand the basic concepts of object oriented programming languages.
3) To learn the techniques of software development in C++.

CONTENTS:

Unit I Object-Oriented Programming Concepts: Introduction, comparison between procedural programming paradigm and object-oriented programming paradigm, basic concepts of object-oriented programming-concepts of an object and a class, interface and implementation of a class, operations on objects, relationship among objects, abstraction, encapsulation, data hiding, inheritance, overloading, polymorphism, messaging.

Unit II Standard Input/output: Concept of streams, hierarchy of console stream classes, input/output using overloaded operators >> and << and member functions of i/o stream classes, formatting output, formatting using ios class functions and flags, formatting using manipulators.

Unit III Classes and Objects: Specifying a class, creating class objects, accessing class members, access specifiers, static members, use of const keyword, friends of a class, empty classes, nested classes, local classes, abstract classes, container classes, bit fields and classes.

Unit IV Pointers and Dynamic Memory Management: Declaring and initializing pointers, accessing data through pointers, pointer arithmetic, memory allocation (static and dynamic), dynamic memory management using new and delete operators, pointer to an object, this pointer, pointer related problems - dangling/wild pointers, null pointer assignment, memory leak and allocation failures.

Unit V Constructors and Destructors: Need for constructors and destructors, copy constructor, dynamic constructors, explicit constructors, destructors, constructors and destructors with static members, initializer lists.

Unit VI Operator Overloading and Type Conversion: Overloading operators, rules for overloading operators, overloading of various operators, type conversion - basic type to class type, class type to basic type, class type to another class type.

Unit VII Inheritance: Introduction, defining derived classes, forms of inheritance, ambiguity in multiple and multipath inheritance, virtual base class, object slicing, overriding member functions, object composition and delegation, order of execution of constructors and destructors.

Unit VIII Virtual functions & Polymorphism: Concept of binding - early binding and late binding, virtual functions, pure virtual functions, abstract classes, virtual destructors.

Unit IX Exception Handling: Review of traditional error handling, basics of exception handling, exception handling mechanism, throwing mechanism, catching mechanism, rethrowing an exception, specifying exceptions.

Unit X Templates and Generic Programming: Template concepts, Function templates, class templates, illustrative examples.

Unit XI Files: File streams, hierarchy of file stream classes, error handling during file operations, reading/writing of files, accessing records randomly, updating files.

COURSE OUTCOMES:

After competition of course the student will able to:

CO1: Distinguish between top-down and bottom-up programming approach and apply bottom-up approach to design C++ programs.

CO2: Perform programs using C++ features such as composition of objects, operator overloading, constructor and destructor, dynamic memory allocation and type conversion.

CO3: Design programs using concepts of inheritance and polymorphism which relates it to reusability and generic data type for the data type independent programming.

CO4: Interpret and design the exception handling techniques for resolving run-time errors and handle large data set using file I/O.

List of Books Referred.

1. Lafore R., Object Oriented Programming in C++, Waite Group.
2. E. Balagurusamy, Object Oriented Programming with C++, Tata McGraw Hill.
3. R. S. Salaria, Mastering Object-Oriented Programming with C++, Salaria Publishing House.
4. Bjarne Stroustrup, The C++ Programming Language, Addison Wesley.
5. Herbert Schildt, The Complete Reference to C++ Language, McGraw Hill-Osborne.
6. Lippman F. B, C++ Primer, Addison Wesley.
7. Farrell- Object Oriented using C++, Cengage Learning.

DATA STRUCTURES LAB
BTCS-306

COURSE OBJECTIVE

1. To develop skills to design and examine simple linear and non linear data structures
2. To build up the ability to identify and apply the suitable data structure for the given real world problem
3. To achieve knowledge in practical applications of data structures

CONTENTS:

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 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 whose elements are stored in on ascending order and implements the following operations (using separate functions): Insert a new element Delete an existing element Search an element Display all the elements Punjab Technical University B.Tech. Computer Science Engineering (CSE) 12 12
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 traversal of graph using breadth-first search.
10. Program to illustrate the traversal of graph using depth-first search.
11. Program to sort an array of integers in ascending order using bubble sort.
12. Program to sort an array of integers in ascending order using selection sort.
13. Program to sort an array of integers in ascending order using insertion sort.
14. Program to sort an array of integers in ascending order using radix sort.
15. Program to sort an array of integers in ascending order using merge sort.
16. Program to sort an array of integers in ascending order using quick sort.
17. Program to sort an array of integers in ascending order using heap sort.
18. Program to sort an array of integers in ascending order using shell sort.
19. Program to demonstrate the use of linear search to search a given element in an array.
20. Program to demonstrate the use of binary search to search a given element in a sorted array in ascending order.

COURSE OUTCOMES

1. To gain knowledge of elementary data structures such as stacks, queues, linked lists, trees and graphs
2. To design and analyze the time and space efficiency of the data structure
3. To identity the appropriate data structure for given problem
4. To have practical knowledge on the application of data structures
5. To discuss different data structures to represent real world problems
6. To design algorithms to solve the problems.

DIGITAL CIRCUITS & LOGIC DESIGN LAB
BTCS-308

CONTENTS:

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 Adder / Full Adder: Realization using basic and 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.

OBJECT ORIENTED PROGRAMMING USING C++ LAB
BTCS-309

COURSE OBJECTIVES

1. To make the students understand the features of object oriented principles and familiarize them with virtual functions, templates and exception handling.
2. To make the students to develop applications using C++.

CONTENTS

1. [Classes and Objects] Write a program that uses a class where the member functions are defined inside a class.
2. [Classes and Objects] Write a program that uses a class where the member functions are defined outside a class.
3. [Classes and Objects] Write a program to demonstrate the use of static data members.
4. [Classes and Objects] Write a program to demonstrate the use of const data members.
5. [Constructors and Destructors] Write a program to demonstrate the use of zero argument and parameterized constructors.
6. [Constructors and Destructors] Write a program to demonstrate the use of dynamic constructor.
7. [Constructors and Destructors] Write a program to demonstrate the use of explicit constructor.
8. [Initializer Lists] Write a program to demonstrate the use of initializer list.
9. [Operator Overloading] Write a program to demonstrate the overloading of increment and decrement operators.
10. [Operator Overloading] Write a program to demonstrate the overloading of binary arithmetic operators.
11. [Operator Overloading] Write a program to demonstrate the overloading of memory management operators.
12. [Typecasting] Write a program to demonstrate the typecasting of basic type to class type.

13. [Typecasting] Write a program to demonstrate the typecasting of class type to basic type.
14. [Typecasting] Write a program to demonstrate the typecasting of class type to class type.
15. [Inheritance] Write a program to demonstrate the multilevel inheritance.
16. [Inheritance] Write a program to demonstrate the multiple inheritance.
17. [Inheritance] Write a program to demonstrate the virtual derivation of a class.
18. [Polymorphism] Write a program to demonstrate the runtime polymorphism.
19. [Exception Handling] Write a program to demonstrate the exception handling.
20. [Templates and Generic Programming] Write a program to demonstrate the use of function template.
21. [Templates and Generic Programming] Write a program to demonstrate the use of class template.
22. [File Handling] Write a program to copy the contents of a file to another file byte by byte. The name of the source file and destination file should be taken as command-line arguments,
23. [File Handling] Write a program to demonstrate the reading and writing of mixed type of data.
24. [File Handling] Write a program to demonstrate the reading and writing of objects.

COURSE OUTCOMES:

After the completion of the course the student:

1. Will have sufficient knowledge of object oriented programming
2. Will be able to map and solve real world problems using C++ programming
3. Will be able to store and retrieve data from files through C++ programming.
4. Will have thorough knowledge about run time exceptions and they will be able to handle them practically.

Fourth Semester

OPERATING SYSTEM

BTCS-401

COURSE OBJECTIVES:

1. To learn the fundamentals of Operating Systems.
2. To learn the mechanisms of OS to handle processes and threads and their communication
3. To learn the mechanisms involved in memory management in contemporary OS
4. To gain knowledge on distributed operating system concepts that includes architecture, Mutual exclusion algorithms, deadlock detection algorithms and agreement protocols
5. To learn programmatically to implement simple OS mechanisms .

CONTENTS:

1. Introduction to Operating system, Role of Operating System as resource manager, function of kernel and shell, operating system structures, views of an operating system.

2. **Process management:** CPU scheduling, Scheduling Algorithms, PCB, Process synchronization, Deadlocks, Prevention, Detection and Recovery

3. **Memory Management:** Overlays, Memory management policies, Fragmentation and its types, Partitioned memory managements, Paging, Segmentation, Need of Virtual memories, Page replacement Algorithms, Concept of Thrashing

4. **Device Management:** I/O system and secondary storage structure, Device management policies, Role of I/O traffic controller, scheduler

5. **File Management:** File System Architecture, Layered Architecture, Physical and Logical File Systems, Protection and Security: [5]

6. Brief study to multiprocessor and distributed operating systems. [4]

7. **Case Studies:** LINUX / UNIX Operating System and Windows based operating systems. Recent trends in operating system.

COURSE OUTCOMES:

CO1: Analyze the structure of OS and basic architectural components involved in OS design.
CO2: Analyze and design the applications to run in parallel either using process or thread models of different OS.
CO3: Analyze the various device and resource management techniques for timesharing and distributed systems.
CO4: Understand the Mutual exclusion, Deadlock detection and agreement protocols of Distributed operating system.

Suggested Readings/ Books:

1. A Silberschatz and Peter B. Galvin, “Operating System Concepts" Addison Wesley Publishing Company

2. Dhamdhere, ―Systems Programming & Operating Systems” Tata McGraw Hill
3. Gary Nutt, “Operating Systems Concepts”, Pearson Education Ltd. 3rd Edition
4. Operating System by Madnick Donovan
5. Operating System by Stallings
6. Ida M.Flynn Understanding Operating Systems -, Cengage Learning

DISCRETE MATHEMATICS
BTCS-402

COURSE OBJECTIVES:

The objective of this course is to provide the necessary back ground of discrete structures with particular reference to the relationships between discrete structures and their data structure counterparts including algorithm development.

CONTENTS:

1. Sets, relations and functions: Introduction, Combination of Sets, ordered pairs, proofs of general identities of sets, relations, operations on relations, properties of relations and functions, Hashing Functions, equivalence relations, compatibility relations, partial order relations.
2. Rings and Boolean algebra: Rings, Subrings, morphism of rings ideals and quotient rings. Euclidean domains Integral domains and fields Boolean Algebra direct product morphisms Boolean sub-algebra Boolean Rings Application of Boolean algebra (Logic Implications, Logic Gates, Karnaughmap)
3. Combinatorial Mathematics: Basic counting principles Permutations and combinations Inclusion and Exclusion Principle Recurrence relations, Generating Function, Application.
4. Monoids and Groups: Groups Semigroups and monoids Cyclic semigraphs and submonoids, Subgroups and Cosets. Congruence relations on semigroups. Morphisms. Normal subgroups. Dihedral groups.
5. Graph Theory: Graph- Directed and undirected, Eulerian chains and cycles, Hamiltonian chains and cycles Trees, Chromatic number Connectivity, Graph coloring, Plane and connected graphs, Isomorphism and Homomorphism. Applications.

COURSE OUTCOMES

After Successful completion of this course the students will be able to:

CO1. Define and Reason mathematically about basic data types and structures (such as numbers, sets, graphs, and trees) used in computer algorithms and systems

CO2. Analyze sets, relations & functions with operations, and identify their structure. Reason and Conclude properties about the structure based on the observations.
CO3. Identify and Apply properties of combinatorial structures and properties - know the basic techniques in combinatorics and counting.
CO4. Gain the conceptual background needed to be able to identify structures of algebraic nature, and discover, prove and use properties about them.
CO5. Comprehend and Evaluate elementary mathematical arguments, rigorous definitions and conclusions about mathematical models and identify fallacious reasoning and statements.

Use graph theoretic models and data structures to design and solve basic problems in Informatics like network connectivity, etc

List of Suggested Readings/ Books:
1. Discrete Mathematics (Schaum series) by Lipschutz (McGraw Hill).
2. Applied Discrete Structures for Computer Science by Alan Doerr and Kenneth Levarseur.
3. Discrete Mathematics by N Ch SN Iyengar, VM Chandrasekaran.
4. Discrete Mathematics and Graph Theory(Cengage Learning) by Sartha
5. Discrete Mathematics and its Applications. Kenneth H Rosen.(McGraw Hill)
6. Elements of discrete mathematics. C L Liu (McGraw Hill)

COMPUTER NETWORKS-I
BTCS-403

COURSE OBJECTIVES

This course provides knowledge about computer network related hardware and software using a layered architecture.

CONTENTS

1. Introduction to Computer Networks:
Data Communication System and its components, Data Flow, Computer network and its goals, Types of computer networks: LAN, MAN, WAN, Wireless and wired networks, broadcast and point to point networks, Network topologies, Network software: concept of layers, protocols, interfaces and services, SO-OSI reference model, TCP/IP reference model.
2. Physical Layer:
Concept of Analog & Digital Signal, Bandwidth, Transmission Impairments: Attenuation, Distortion, Noise, Data rate limits : Nyquist formula, Shannon Formula, Multiplexing : Frequency Division, Time Division, Wavelength Division, Introduction to Transmission Media : Twisted pair, Coaxial cable, Fiber optics, Wireless transmission (radio, microwave, infrared), Switching: Circuit Switching, Message Switching ,Packet Switching & their comparisons.
3. Data Link Layer:
Design issues, Framing, Error detection and correction codes: checksum, CRC, hamming code, Data link protocols for noisy and noiseless channels, Sliding Window Protocols: Stop & Wait ARQ, Go-back-N ARQ, Selective repeat ARQ, Data link protocols: HDLC and PPP.
4. Medium Access Sub-Layer:

Static and dynamic channel allocation, Random Access: ALOHA, CSMA protocols, Controlled Access: Polling, Token Passing, IEEE 802.3 frame format, Ethernet cabling, Manchester encoding, collision detection in 802.3, Binary exponential back off algorithm.

5. Network Layer:

Design issues, IPv4 classful and classless addressing, subnetting, Routing algorithms: distance vector and link state routing, Congestion control: Principles of Congestion Control, Congestion prevention policies, Leaky bucket and token bucket algorithms.

6. Transport Layer:

Elements of transport protocols: addressing, connection establishment and release, flow control and buffering, multiplexing and de-multiplexing, crash recovery, introduction to TCP/UDP protocols and their comparison.

7. Application Layer:

World Wide Web (WWW), Domain Name System (DNS), E-mail, File Transfer Protocol (FTP), Introduction to Network security.

COURSE OUTCOMES:

CO1. Understand the basic of data communication and data transmission.
CO2. Know about the working of computer networks, various topologies and their use in real word application
CO3. Study all seven layers related to OSI model with complete structure, function and their role in data communication.
CO4. A comprehensive analysis of security aspect related to computer networks.

List of Books Referred :

1. Computer Networks, 4th Edition, Pearson Education by Andrew S. Tanenbaum
2. Data Communication & Networking, 4th Edition, Tata McGraw Hill. By Behrouz A. Forouzan.
3. Computer Networking, 3rd Edition, Pearson Education by James F. Kurose and Keith W. Ross
4. Internetworking with TCP/IP, **Volume-I**, Prentice Hall, India by Douglas E. Comer.

MICROPROCESSORS AND ASSEMBLY LANGUAGE PROGRAMMING
BTCS-404

COURSE OBJECTIVES: The course is intended to give students good understanding of internal architectural details and functioning of microprocessors.

CONTENTS:

1. Introduction: Introduction to Microprocessors, history, classification, recent microprocessors.
2. Microprocessor Architecture: 8085 microprocessor Architecture. Bus structure, I/O, Memory & Instruction execution sequence & Data Flow, Instruction cycle. System buses, concept of address Bus, Data Bus & Control Bus, Synchronous & Asynchronous buses.
3. I/O memory interface: Data transfer modes: Programmable, interrupt initiated and DMA. Serial & parallel interface, Detail study of 8251 I/O Processor & 8255 programmable peripheral

interfaces.
4. Instruction set & Assembly Languages Programming: Introduction, instruction & data formats, addressing modes, status flags, 8085 instructions, Data transfer operations, Arithmetic operations, Logical operations, Branch operations.
5. Case structure & Microprocessor application: Interfacing of keyboards and seven segment LED display, Microprocessor controlled temperature system (MCTS), Study of traffic light system, stepper motor controller, Microprocessor based micro computers.
6. Basic architecture of higher order microprocessors: Basic introduction to 8086 family, Motorola 68000, Pentium processors.

COURSE OUTCOMES:

At end of the course, students will be able to:

1. Describe the general architecture of a microcomputer system and architecture &organization of 8085 and understand the difference between 8085 & 8086 Microprocessor and advanced microprocessors.
2. Understand and classify the instruction set of 8085 microprocessor and distinguish the use of different instructions and apply it in assembly language programming.
3. Understand and realize the Interfacing of various I/O devices with 8085 microprocessor

Suggested Readings/ Books:

1. Ramesh Gaonkar, "8085 Microprocessor ",PHI Publications.
2. Daniel Tabak, "Advanced Microprocessors", McGraw- Hill, Inc., Second Edition 1995.
3. Douglas V. Hall, " Microprocessors and Interfacing: Programming and Hardware", Tata McGraw Hill Edition,1986.
4. Charles M.Gilmore," Microprocessors: Principles and Applications", McGraw Hill.
5. Ayala Kenneth, "The 8086 Microprocessor Programming and Interfacing", Cengage Learning

SYSTEM PROGRAMMING

BTCS-405

COURSE OBJECTIVES: This course provides knowledge to design various system programs.

CONTENTS:

1. Introduction: Introduction to system programming and different types of system programs – editors, assemblers, macro-processors, compilers, linkers, loader, debuggers.

2. Assemblers: Description of single pass and two pass assemblers, use of data structures like OPTAB and SYMTAB, etc.

3. Macroprocessors: Description of macros, macro expansion, conditional and recursive macro expansion.

4. Compilers: Various phases of compiler – lexical, syntax and semantic analysis, intermediate code generation, code optimization techniques, code generation, Case study: LEX and YACC.

5. Linkers and Loaders: Concept of linking, different linking schemes, concept of loading and various loading schemes.

6. Editors: Line editor, full screen editor and multi window editor, Case study MS-Word, DOS

Editor and vi editor.

7. Debuggers: Description of various debugging techniques.

COURSE OUTCOMES:

Students will be able:

1) To understand the basics of system programs like editors, compiler, assembler, linker, loader, interpreter and debugger.
2) Describe the various concepts of assemblers and macroprocessors.
3) To understand the various phases of compiler and compare its working with assembler.
4) To understand how linker and loader create an executable program from an object module created by assembler and compiler.
5) To know various editors and debugging techniques.

Suggested Readings/ Books:

1. Donovan J.J., "Systems Programming", New York, Mc-Graw Hill, 1972.
2. Dhamdhere, D.M., "Introduction to Systems Software", Tata Mc-Graw Hill, 1996.
3. Aho A.V. and J.D. Ullman ,"Principles of compiler Design" Addison Wesley/ Narosa 1985.
4. Kenneth C. Louden," Compiler Construction", Cengage Learning.

OPERATING SYSTEM LAB

BTCS-406

COURSE OBJECTIVES

1. To make students able to learn different types of operating systems along with concept of file systems and CPU scheduling algorithms used in operating system.
2. To provide students knowledge of memory management and deadlock handling algorithms.
3. At the end of the course, students will be able to implement various algorithms required for management, scheduling, allocation and communication used in operating system.

CONTENTS:

1. Installation Process of various operating systems

2. Virtualization, Installation of Virtual Machine Software and installation of Operating System on Virtual Machine

3. Commands for files & directories: cd, ls, cp, md, rm, mkdir, rmdir. Creating and viewing files using cat. File comparisons. Disk related commands: checking disk free spaces. Processes in linux, connecting processes with pipes, background processing, managing multiple processes. Manual help. Background process: changing process priority, scheduling of processes at command, batch commands, kill, ps, who, sleep. Printing commands, grep, fgrep, find, sort, cal, banner, touch, file. File related commands ws, sat, cut, grep.

4. 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.

COURSE OUTCOMES

Upon the completion of Operating Systems practical course, the student will be able to:

1. Understand and implement basic services and functionalities of the operating system using system calls.
2. Use modern operating system calls and synchronization libraries in software/ hardware interfaces.
3. Understand the benefits of thread over process and implement synchronized programs using multithreading concepts.
4. Analyze and simulate CPU Scheduling Algorithms like FCFS, Round Robin, SJF, and Priority.
5. Implement memory management schemes and page replacement schemes.

6. Simulate file allocation and organization techniques.

7. Understand the concepts of deadlock in operating systems and implement them in multiprogramming system.

COMPUTER NETWORKS-I LAB
BTCS-407

COURSE OBJECTIVES

This course provides knowledge about computer network related hardware and software using a layered architecture.

CONTENTS

1. Write specifications of latest desktops and laptops.
2. Familiarization with Networking Components and devices: LAN Adapters, Hubs, Switches, Routers etc.
3. Familiarization with Transmission media and Tools: Co-axial cable, UTP Cable, Crimping Tool, Connectors etc.
4. Preparing straight and cross cables.

5. Study of various LAN tpologies and their creation using network devices, cables and computers.
6. Configuration of TCP/IP Protocols in Windows and Linux.
7. Implementation of file and printer sharing.
8. Designing and implementing Class A, B, C Networks
9. Subnet planning and its implementation
10. Installation of ftp server and client.

COURSE OUTCOMES:
CO1. Understanding of basic computer network architecture implementation.
CO2. Installation of various components, devices and with different modes of Transmission.
CO3. Designing and implementation of IPV4 addressing scheme in wired network.
CO4. Planning and implementation of subnet with classful addressing scheme.
CO5. Installation of various servers.

MICROPROCESSOR AND ASSEMBLY LANGUAGE PROGRAMMING LAB
BTCS-408

COURSE OBJECTIVES

To enable the students to simulate and test the Analog, Digital and mixed Electronics circuits using MATLAB/CASPOC/OrCAD Softwares. To provide a platform for the students to do multidisciplinary projects.

CONTENTS:
1. Introduction to 8085 kit.
2. Addition of two 8 bit numbers, sum 8 bit.
3. Subtraction of two 8 bit numbers.
4. Find 1's complement of 8 bit number.
5. Find 2's complement of 8 bit number.
6. Shift an 8 bit no. by one bit.
7. Find Largest of two 8 bit numbers.
8. Find Largest among an array of ten numbers (8 bit).
9. Sum of series of 8 bit numbers.
10. Introduction to 8086 kit.
11. Addition of two 16 bit numbers, sum 16 bit.
12. Subtraction of two 16 bit numbers.
13. Find 1's complement of 16 bit number.
14. Find 2's complement of 16 bit number.

COURSE OUTCOMES
After studying this course the students will be able to:
1. Solve basic binary math operations using the instructions of microprocessor 8085.
2. Apply programming knowledge using the capabilities of the stack, the program counter
3. Design, code and debugs Assembly Language programs to implement simple programs
4. Execute a machine code program on the training boards.

SYSTEM PROGRAMMING LAB

BTCS-409

COURSE OBJECTIVES: This course provides knowledge to design various system programs.

CONTENTS:

1. Create a menu driven interface for

a) Displaying contents of a file page wise

b) Counting vowels, characters, and lines in a file.

c) Copying a file

2. Write a program to check balance parenthesis of a given program. Also generate the error report.

3. Write a program to create symbol table for a given assembly language program.

4. Write a program to create symbol table for a given high-level language program.

5. Implementation of single pass assembler on a limited set of instructions.

6. Exploring various features of debug command.

7. Use of LAX and YACC tools.

COURSE OUTCOMES

After the completion of this course the students will be able to:
1. Study the architecture of a hypothetical machine, its assembly language, macro language
2. Program in assembly language.
3. Understand the structure and design of assemblers, linkers and loaders.
4. Understand the concepts and theory behind the implementation of high level programming languages.
5. Use tools like LEX & YACC.

Fifth Semester

COMPUTER NETWORKS-II

BTCS-501

COURSE OBJECTIVES

1. To have good understanding of the concepts of network security, wireless, Adhoc and various emerging network technologies.
2. To familiarize the students with standards and design aspects of wireless communication systems.

CONTENTS

1. **Network Security:** Fundamentals of network security, Basics of IPv6, IPsec: overview of IPsec, IP and IPv6, Authentication header (AH), Encapsulating Security Payload (ESP).
2. **Internet Key Exchange (IKE**): History, Photuris, Simple Key-management for Internet protocols (SKIP), IKE phases, IKE encoding.
3. **Adhoc networks**: Features, advantages and applications, Adhoc versus Cellular networks, Network architecture, Protocols: MAC protocols, Routing protocols, Technologies.
4. **Wireless Communication Systems**: Evolution, examples of wireless communication systems, 2G Cellular networks, Evolution for 2.5G TDMA Standards, IS-95B for 2.5G CDMA.
5. **3G wireless networks**: wireless local loop (WLL), Local Multipoint Distribution System (LMDS), Wireless local Area Networks (WLANs), Bluetooth and Personal Area Networks.
6. **Wireless System Design:** Introduction, Frequency reuse, channel assignment strategies, handoff strategies, interference and system capacity, improving coverage and capacity in cellular systems.

COURSE OUTCOMES

CO1. State the fundamentals related to network security and basics of IPv6 and IPsec.
CO2.State the fundamentals related to network security and basics of IPv6 and IPsec.
CO3. Explain various protocols related to internet key exchange.
CO4. Study Ad Hoc Networks and its protocols.
CO5. Define various examples of wireless communication system, standards related to 2G and 3G wireless networks.

CO6. Design wireless mobile network according to parameters such as frequency reuse, handoff strategies and system capacity.

List of Recommended Books & Title

1.Theodore S. Rappaport, Wireless Communication: Principles and Practices (2nd Edition), Pearson Education.
2. Charlie Kaufman, Radio Perlman, Mike Speciner, Neywork security, 2nd ed., PHI.
3. Sunilkumar S. Manvi, Mahabaleshwar S. Kakkasageri, Wireless and mobile networks: concepts and protocols, Wiley India.
4. Michael A. Gallo & William M. Hancock, "Computer Communications and Networking Technologies", Cengage Learning / Thomson Brooks / Cole
5. S. Keshav, "An Engineering Approach to Computer Networking", Pearson Education.
6. Mayank Dave, "Computer Networks", Cengage Learning

RELATIONAL DATABASE MANAGEMENT SYSTEM

BTCS-502

COURSE OBJECTIVES

1. To introduce the concept of Database Management System and its applications in real world domain.
2. To familiarize the students with Relational Database Management System concepts.
3. To acquaint the students with concepts of Normalisation, Transaction & Concurrency control.
4. To train the students with Structure Query Language and its use in Database Management.

CONTENTS:

Introduction to Database Systems: File Systems Versus a DBMS, Advantages of a DBMS, Describing and Storing Data in a DBMS, Database System Architecture, DBMS Layers, Data independence.

Physical Data Organization: File Organization and Indexing, Index Data Structures, Hashing, B-trees, Clustered Index, Sparse Index, Dense Index, Fixed length and Variable Length Records.

Data Models: Relational Model, Network Model, Hierarchical Model, ER Model: Entities, Attributes and Entity Sets, Relationships and Relationship Sets, Constraints, Weak Entities, Class Hierarchies, Aggregation, Conceptual Database Design with the ER Model, Comparison of Models.

The Relational Model: Introduction to the Relational Model, ER to Relational Model Conversion, Integrity Constraints over Relations, Enforcing Integrity Constraints, Relational Algebra, Relational Calculus, Querying Relational Data.

Relational Query Languages: SQL: Basic SQL Query, Creating Table and Views, SQL as DML, DDL and DCL, SQL Algebraic Operations, Nested Queries, Aggregate Operations, Cursors, Dynamic SQL, Integrity Constraints in SQL, Triggers and Active Database, Relational Completeness, Basic Query Optimization Strategies, Algebraic Manipulation and Equivalences.

Database Design: Functional Dependencies, Reasoning about Functional Dependencies, Normal Forms, Schema Refinement, First, Second and Third Normal Forms, BCNF, Multi-valued Dependency, Join Dependency, Fourth and Fifth Normal Forms, Domain Key Normal Forms, Decompositions.

Transaction Management: ACID Properties, Serializability, Two-phase Commit Protocol, Concurrency Control, Lock Management, Lost Update Problem, Inconsistent Read Problem , Read-Write Locks, Deadlocks Handling, 2PL protocol.

Database Protection: Threats, Access Control Mechanisms, Discretionary Access Control, Grant and Revoke, Mandatory Access Control, Bell LaPadula Model, Role Based Security, Firewalls, Encryption and Digital Signatures.

COURSE OUTCOMES:

After the completion of the course the student will be able:

CO1. To understand the concepts and architecture of Database Management System and distinguish it from RDBMS.

CO2. To conceptualize, design and normalize database and models it with the use of appropriate data types for storage in databases.

CO3. To study and implement data base with the help of Structured Query Language on platforms like ORACLE.

CO4. To understand Concurrency Control techniques and security aspects in databases.

List Of Recommended Books And Titles

1. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, Database S ystem Concepts, Tata McGraw- Hill.
2. RamezElmasri, ShamkantNavathe, Fundamentals of Database Systems, Fifth Edition, Pearson.
3. C.J. Date, An Introduction to Database Systems, Eighth Edition, Pearson Education.
4. S. K. Singh, Database Systems Concepts, Design and Applications, Pearson Education.
5. Raghu Ramakrishnan, Johannes Gehrke, Database Management Systems, Tata McGraw-Hill.

DESIGN & ANALYSIS OF ALGORITHMS

BTCS-503

COURSE OBJECTIVES

- Analyze the asymptotic performance of algorithms
- Demonstrate a familiarity with major algorithms and data structures
- Familiarizing students with specific algorithms for a number of important computational problems like sorting, searching, and graphs, …etc,
- Introducing the concept of NP-complete problems and different techniques to deal with them.

CONTENTS

- **Introduction.** What is an algorithm? Time and space complexity of an algorithm. Comparing the performance of different algorithms for the same problem. Different orders of growth. Asymptotic notation. Polynomial vs. Exponential running time.
- **Basic Algorithm Design Techniques.** Divide-and-conquer, greedy, randomization, and dynamic programming. Example problems and algorithms illustrating the use of these techniques.
- **Graph Algorithms**. Graph traversal: breadth-first search (BFS) and depth-first search (DFS). Applications of BFS and DFS. Topological sort. Shortest paths in graphs: Dijkstra and Bellman-Ford. Minimum spanning trees.
- **Sorting and searching**. Binary search in an ordered array. Sorting algorithms such as Merge sort, Quick sort, Heap sort, Radix Sort, and Bubble sort with analysis of their running times. Lower bound on sorting.
- **Median and order statistics.** NP-completeness. Definition of class NP. NP-hard and NP-complete problems. 3SAT is NP-complete. Proving a problem to be NP-complete using polynomial-time reductions. Examples of NP-complete problems. Coping with NP-completeness. Approximation algorithms for various NP-complete problems.
- **Advanced topics.** Pattern matching algorithms: Knuth-Morris-Pratt algorithm. Algorithms in Computational Geometry: Convex hulls. Fast Fourier Transform (FFT) and its applications. Integer and polynomial arithmetic. Matrix multiplication :Strassen's algorithm.

COURSE OUTCOMES

CO1. Analyze worst-case running times of algorithms using asymptotic analysis.

CO2. Describe the Different paradigms and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize different paradigm programming algorithms, and analyze them.

CO3. Explain what competitive analysis is and to which situations it applies. Perform competitive analysis

CO4. Analyze randomized algorithms. Employ indicator random variables and linearity of expectation to perform the analyses. Recite analyses of algorithms that employ this method of analysis.

CO5. Explain what an approximation algorithm is, and the benefit of using approximation algorithms. Analyze the approximation factor of an algorithm.

List Of Recommended Books And Titles:

1. Algorithm Design by J. Kleinberg and E. Tardos.
2. Introduction to Algorithms by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein.
3. Algorithms by S. Dasgupta, C.H. Papadimitriou, and U.V. Vazirani.
4. Algorithm Design: Foundations, Analysis, and Internet Examples by Michael T. Goodrich and Roberto Tamassia.
5. The Design and Analysis of Computer Algorithms by A. V. Aho, J. E. Hopcroft, and J. D. Ullman.
6. The Art of Computer Programming, Volumes 1, 2, and 3, by Donald Knuth.

COMPUTER GRAPHICS
BTCS-504

COURSE OBJECTIVES

Understanding the fundamental graphical operations and the implementation on computer, Get a glimpse of recent advances in computer graphics, Understanding user interface issues that make the computer easy for the novice to use.

CONTENTS

1. Introduction: Computer Graphics and its applications, Elements of a Graphics, Graphics Systems: Video Display Devices, Raster Scan Systems, Random Scan Systems, Input devices.
2. Basic Raster Graphics: Scan conversion- Point plot technique, Line drawing, Circle generating and Ellipse generating algorithms.
3. Two-dimensional Geometric Transformations : Basic Transformations-Translation, Rotation and Scalling, Matrix Representation and Homogeneous Coordinates, Composite Transformations, Reflection and Shearing transformations.
4. Clipping: Window to viewport transformation, Clipping Operations- Point Clipping, Line Clipping, Polygon Clipping and Text Clipping.
5. Filling Techniques: Scan line algorithms, Boundary-fill algorithm, Flood-fill algorithm, Edge fill and fence fill algorithms,

6. Elementary 3D Graphics: Plane projections and its types, Vanishing points, Specification of a 3D view.
7. Visibility: Image and object precision, Hidden edge/surface removal or visible edge/surface determination techniques; z buffer algorithms, Depth sort algorithm, Scan line algorithm and Floating horizon technique.
8. Advance Topics: Introduction of Rendering, Raytracing, Antialiasing, Fractals, Gourard and Phong shading.

COURSE OUTCOMES
Students will be able:

CO1.To tell the application domains of Computer Graphics and various devices associated with it.
CO2. To illustrate raster graphics techniques using algorithms and numerica.
CO3. To elaborate two-dimensional geometric transformations.
CO4. To fathom the concept of clipping and filling using different algorithms.
CO5. To explain elementary three dimensional graphics, visibility and other advanced topics like rendering and ray tracing using various examples.

List Of Recommended Books And Titles:

1. Donald Hearn and M.Pauline Baker, "Computer Graphics", Second Edition, PHI/Pearson Education.
2. Zhigandxiang, Roy Plastock, Schaum's outlines, "Computer Graphics Second Edition", Tata Mc-Grawhill edition.
3. C, Foley, VanDam, Feiner and Hughes, "Computer Graphics Principles & Practice", Second Edition, Pearson Education

COMPUTER PERIPHERALS & INTERFACES
BTCS-505

COURSE OBJECTIVE

1. To learn the functional and operational details of various peripheral devices.
2. To have sufficient knowledge of computer hardware equipment as well as interfaces.

CONTENTS

1. **SYSTEM RESOURCES**: Interrupt, DMA Channel, I/O Port Addresses and resolving and resolving the conflict of resources. I/O buses- ISA, EISA, Local bus, VESA Local bus, PCI bus, PCI Express, Accelerated graphics port bus.

2. **IDE & SCSI Interfaces**: IDE origin, IDE Interface ATA standards ATA1 to ATA7. ATA feature, ATA RAID and SCSI RAID, SCSI Cable and pin Connector pin outs SCSI V/s IDE Advantages and limitation.

3. **Video Hardware**: Video display technologies, DVI Digital signals for CRT Monitor, LCD Panels, Video adapter types, Integrated Video/ Motherboard chipset, Video RAM, Video driver and multiple Monitor, Graphic accelerators. Advanced 3D Technologies, TV Tuner and Video Capture upgrades troubleshooting Video Cards and Drivers.

4. **I/O Interfaces**: I/O Interfaces from USB and IEEE1394, I/O Interface from serial and Parallel to IEEE1394 and USB 961, Parallel to SCSI converter. Testing of serial and parallel port, USB Mouse/ Keyboard Interfaces.

5. **Input/ Output Driver software aspects**: Role of device driver DOS and UNIX/ LINUX device drivers.

6. Design & Integration of Peripheral devices to a computer system as a Case Study.

7. **Future Trends**: Detailed Analysis of recent Progress in the Peripheral and Bus systems. Some aspects of cost Performance analysis while designing the system.

COURSE OUTCOMES

Students will be able to

CO1. Familiarize with the operation of a sophisticated computer system, including high-performance peripheral interfaces.

CO2. Analyze digital interfaces IDE, ATA, SCSI.

CO3. Select appropriate and compatible computer/peripherals combinations.

CO4. Design digital interface circuits for Input, Output, Keyboard and Display Circuit.

List Of Recommended Books And Titles

1. Douglas V. Hall, "Microprocessors and Interfacing", Tata McGraw Hill 2006.
2. Barry B. Brey&C.R.Sarma" The intel microprocessors," Pearson 2003.
3. P. Pal Chandhari , "Computer Organization and design" Prentice Hall of India Pvt. Ltd, 1994.
4. Del Corso, H.Kirrman, JD Nicond, "Microcomputer buses & links" Academic Press 1986.

RELATIONAL DATABASE MANAGEMENT SYSTEM LAB
BTCS-506

COURSE OBJECTIVES

1. To create a database and query it using SQL.

2. Understand the significance of integrity constraints, referential integrity constraints, triggers.
3. To provide a strong formal foundation in database concepts and technologies.

CONTENTS:

1. Introduction to SQL and installation of SQL Server / Oracle.

2. Data Types, Creating Tables, Retrieval of Rows using Select Statement, Conditional Retrieval of Rows, Alter and Drop Statements.

3. Working with Null Values, Matching a Pattern from a Table, Ordering the Result of a Query, Aggregate Functions, Grouping the Result of a Query, Update and Delete Statements.

4. Set Operators, Nested Queries, Joins, and Sequences.

5. Views, Indexes, Database Security and Privileges: Grant and Revoke Commands, Commit and Rollback Commands.

6. PL/SQL Architecture, Assignments and Expressions, Writing PL/SQL Code, Referencing Non-SQL parameters.

7. Stored Procedures and Exception Handling.

8. Triggers and Cursor Management in PL/SQL.

COURSE OUTCOMES

After the completion of the course the student will be able:

CO1. To perform different operations on Relational databases using SQL.

CO2. To secure database by providing/revoking Privileges.

CO3. To make PL/SQL programs including stored Procedures, Cursors, Triggers.

List Of Recommended Tools

Suggested Tools – MySQL, DB2, Oracle, SQL Server 2012, Postgre SQL, SQL lite.

COMPUTER NETWORKS –II LAB
BTCS 507

COURSE OBJECTIVES

To familiarize the students with LAN configuration based on IPv4 & IPv6 and to have the insight of simulations for Ad Hoc Networks

CONTENTS

1. To configure the IP address for a computer connected to LAN and to configure network parameters of a web browser for the same computer.

2. To plan IPv6 address scheme for a local area network comprising of 'n' terminals.

3. To develop programs for implementing / simulating routing algorithms for Adhoc networks.

4. To install any one open source packet capture software like wireshark etc.

5. To configure Wireless Local Loop.

6. To plan Personal Area Network.

7. To configure WLAN.

8. To configure Adhoc networks.

9. To install and configure wireless access points

COURSE OUTCOMES

CO1. Configure a LAN based on IPv4 address scheme and understand and implement IPv6 address scheme for a LAN.

CO2. Configure and simulate any scenario of an Adhoc network and analyze various parameters related with their study.

CO3. Devise and design a system to capture and analyze the incoming traffic using packet capturing software package.

CO4. Configure WLL, PAN's, WLANS and wireless access points.

DESIGN & ANALYSIS OF ALGORITHMS LAB
BTCS-508

COURSE OBJECTIVES

In this laboratory after completing experiments student has to learn how to analyze a problem & design the solution for the problem. In addition to that, solution must be optimum, i.e., time complexity & memory usage of the solution must be very low.

CONTENTS

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 shortest paths in a graph with positive edge weights using Dijkstra's algorithm.

13. Code and analyze to find shortest paths in a graph with arbitrary edge weights using Bellman-Ford algorithm.

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

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

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

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

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

COURSE OUTCOMES:

The students should be able to:

Design algorithms using appropriate design techniques (brute-force, greedy, dynamic programming, etc.)

Implement a variety of algorithms such as sorting, graph related, combinatorial etc., in a high level language.

Analyze and compare the performance of algorithms using language features.

Apply and implement learned algorithm design techniques and data structures to solve real world problems.

COMPUTER GRAPHICS LAB
BTCS-509

COURSE OBJECTIVES

To collaborate the concepts of C++ and computer graphics to implement various line, circle, and ellipse generating algorithms. To apprehend the implementation of certain transformation and clipping techniques.

CONTENTS:

1. To plot a point (pixel) on the screen.
2. To draw a straight line using DDA Algorithm.
3. To draw a straight line using Bresenham's Algorithm.
4. Implementation of mid-point circle generating Algorithm.
5. Implementation of ellipse generating Algorithm.
6. To translate an object with translation parameters in X and Y directions.
7. To scale an object with scaling factors along X and Y directions.
8. To rotate an object with a certain angle about origin.
9. Perform the rotation of an object with certain angle about an arbitrary point.
10. To perform composite transformations of an object.
11. To perform the reflection of an object about major axis.

12. To clip line segments against windows using Cohen Sutherland Algorithm.
13. Perform the polygon clipping against windows using Sutherland Hodgeman technique.
14. Fill a rectangle with a specified color using scan line algorithm.
15. Implementation of flood-fill and boundary-fill algorithms.

COURSE OUTCOMES

Students will be able to:
CO1. Comprehend the use of C++ programming code to implement different line, circle and ellipse generating algorithms
CO2. Implement various transformation schemes comprising rotation, scaling and translation
CO3. Fathom composite transformations of an object
CO4. Apprehend and implement clipping, color filling using C++

Sixth Semester

SIMULATION AND MODELING

BTCS-601

COURSE OBJECTIVES

1. In this course you will study the representation and simulation of physical systems using a range of mathematical formulations.
2. There are many modeling techniques to describe system characteristics. You will learn to develop typical mathematical models.
3. Case studies and engineering software applications are used to illustrate a variety of modeling techniques. Once the models are validated, you can utilize them to predict the behavior of common industrial and engineering systems including: mechanical, electrical, civil, environmental, fluid, magnetic, thermal and transport. Problem solving with these systems may involve graphical, algebraic, numerical, state space, simulation and computational processes.

CONTENTS:

Module1*:* Introduction- When simulation is appropriate and when not, advantages and disadvantages of simulation, application areas in communication, computer and software design, systems and system environment, components of a system, discrete and continuous systems, model of a system, types of models, discrete-event simulation, steps in a simulation study. Simulation Examples- Simulation of queuing systems, on-demand and inventory systems, simulation for reliability analysis etc

Module 2: General Principles- Concepts in discrete event simulation: event scheduling/time advances algorithms, world views. List Processing: properties and operations, data structures and dynamic allocation, techniques;

Module 3: Simulation Software- Integrated environments. Examples and review of some existing software popular and useful in the industry, e.g., Arena, AutoMod, Extend, Flexsim, Micro Saint, ProModel, Quest, SIMUL8, WITNESS etc. Simulation using languages and environments like C++/Java/GPSS/SSF etc. Experimentation and Statistical-Analysis Tools: common features and relevant current products.

Module 4: Statistical Models in Simulation- Terms and concepts. Statistical Models. Review of discrete and continuous distributions. Review of Poisson (stationary and non-stationary) processes, Empirical Distributions, Elementary Queuing Theory- Basic Structure of Queuing Models, Input Source Calling Population), Queue, Queue Discipline, Service Mechanisms. Notations and relationships between *L*, *W*, *Lq*, and *Wq*. Little's Formula, Role of Exponential Distribution and Properties. Birth and Death Processes, M/M/s queues. Finite queue variation in M/M/s/K models with different s values. Finite Calling Population cases, Queuing Models involving Non-Exponential Distributions: M/G/1, M/D/s, M/Ek/s (involving Erlang distribution), Models without a Poisson Input, Models involving hyper exponential distributions, Priority Discipline Queuing Models: Preemptive and Non- Preemptive with results, properties and server number variations, Queuing Networks: Equivalence Property. Infinite Queues in Series and Product Form Solutions. Jackson Networks,

Module 5: Application of Queuing Models- Review of Characteristics (calling population system capacity, arrival processes, behavior and disciplines, service times and mechanisms etc) and notations, Application of Long-Run Measures of Performance: Time average in system, average time spent per customer, Little's Formula and server utilization, and costs. Steady State behavior of Infinite (M/G/1, M/M/c/infinity, M/M/c/N/infinity) and finite (M/M/c/K/K) Calling Population Models, Use of Network of Queues.

Module 6: Random Number Generation- Properties. Generation of Pseudo-Random Numbers, Techniques for Generation of Pseudo-Random Numbers: Linear Congruential, Combined Linear Congruential, Random Number Streams. Tests for Random Numbers: Frequency Tests and Tests for Autocorrelation. Random Variate Generation- Inverse Transform Techniques for Exponential, Uniform, Weibull, Triangular and for Empirical Continuous Distributions. Acceptance-Rejection

Techniques for Poisson (Stationary and Non- Stationary) Distribution and Gamma Distribution. Special Properties like the Direct Transformation for the Normal and Lognormal Distributions, Convolution Method and others.

Module 7: Input Modeling- Data collection, Identifying the Distribution with Data: Histograms, Selection of the Appropriate Family of Distributions, Quantile-Quantile Plots.100 Parameter Estimation: Sample Mean and Sample Variance and various biased and unbiased Estimators. Goodness of Fit Tests applied to Simulation inputs: Chi-Square and Chi-Square with Equal Probabilities, Kolmogorov-Smirnov Tests, p-Values and Best Fits. Verification and Validation of Simulation Models- Verification and Validation of Simulation Models. Calibration and Validation: Face Validity, Validation of Assumptions, Input-Out Transformation Validation.

Module 8: Output Analysis of a Single Model- Output analysis and types of simulation. Stochastic Nature of the Output Data, Measures of Performance and Estimation: Point Estimation and Confidence-Interval Estimation, Output Analysis for Terminating Simulations and Estimation of Probabilities. Output Analysis of Steady State Simulations: Initialization Bias, Error Estimation, Replications, Sample Size and Batch Means for Interval Estimation.

Module 9: Comparison and Evaluation of Alternative System Designs- Comparison of Two System Designs, Sampling with Equal and Unequal Variances. Common Random Numbers, Confidence Intervals with Specified Precision. Comparison of Several System Designs: Bonferroni Approaches to Multiple Comparisons and to Screening and to Selection of the Best. Metamodeling L Sample Linear Regression, Testing for Significance, Multiple Linear Regressions, Random Number Assignment for Regression, Optimization via Simulation: Robust Heuristics.

Module 10: Simulation of Computer Systems- Simulation Tools: Process Orientation and Event Orientation. Model Input: Modulated Poisson Process and Virtual-Memory Referencing. High-Level Simulation, CPU and Memory Simulations, Simulation of Computer Networks- Traffic Modeling, Media Access Control: Token- Passing Protocols and Ethernet, Data Link Layer, TCP,

Model Construction. Simulation Languages: Basic Introduction to Special Simulation Languages:- GPSS/ MATLAB/ Network Simulators.

COURSE OUTCOMES:

On successful completion of this course, the students should be able to:

CO1. Characterize a given engineering system in terms of its essential elements, that is, purpose, parameters, constraints, performance requirements, subsystems, interconnections and environmental context.

CO2. Develop a modeling strategy for a real world engineering system, which considers prediction and evaluation against design criteria, and integrates any required sub-system models

CO3. Assess and select a model for an engineering system taking into consideration its suitability to facilitate engineering decision making and predicted advantages over alternative models.

CO4. Interpret the simulation results of an engineering system model, within the context of its capabilities and limitations, to address critical issues in an engineering project

CO5. Analyze output data produced by a model and test validity of the model

Suggested Readings/ Books:

1. Jerry Banks, John S. Carson II, Barry L. Nelson and David M. Nicol, Discrete-Event System and

Simulation, Prentice Hall of India, New Delhi, 2005

2. Averill M. Law, Simulation modeling and analysis (SIE), Tata McGraw Hill India, 2007

3. David Cloud, Larry Rainey, Applied Modeling and Simulation, Tata McGraw Hill, India.

4. Gabriel A. Wainer, Discrete-event modeling and simulation: a practitioner's approach, CRC Press,

2009.

5. Bernard P. Zeigler, Herbert Praehofer, Tag Gon Kim, Theory of modeling and simulation integrating discrete event and continuous complex dynamic systems, Academic Press, 2000.

6. Walter J. Karplus, George A. Bekey, Boris Yakob Kogan, Modeling and simulation: theory and practice, Springer, 2003.

7. Stanislaw Raczynski, Modeling and simulation: the computer science of illusion, Wiley, 2006.

8. Mohammad Salameh Obaidat, Georgios I. Papadimitriou, Applied system simulation: methodologies and application, Springer, 2003.

9. van Dijk, Nico M.; Boucherie, Richard J. (Eds.) 2011, Queuing Networks: A Fundamental Approach, 798 pp: 148 illus. Springer.

10. Bhat, U. Narayan, an Introduction to Queuing Theory: Modeling and Analysis in Applications Springer 2008 (Birkhäuser Boston).

11. James J. Nutaro, Building software for simulation: theory and algorithms, with applications in C++ Wiley 2010.

RELATIONAL DATABASE MANAGEMENT SYSTEM –II

BTCS-602

COURSE OBJECTIVES

1. To study the basic database concepts of Database Architecture, Normalization, Transaction Processing and Concurrency Control.
2. To familiarize the students with advanced database concepts such as Query Processing, Evaluation and Optimization.
3. To familiarize the students with implementation of various database concepts in object oriented and distributed databases.
4. To enlighten the students with latest emerging database applications and technologies such as OLTP, database mining and Data ware housing.

CONTENTS

Introduction to Database Systems: Database System Concepts and Architecture, Data Models, Data Independence, SQL: DDL, DML, DCL, Normalization: 1NF, 2NF, 3NF, BCNF, 4NF, 5NF.

Query Processing and Optimization: Query Processing, Syntax Analyzer, Query Decomposition, Query Optimization, Heuristic Query Optimization, Cost Estimation, Cost Functions for Select, Join, Query Evaluation Plans.

Transaction Processing and Concurrency Control: Transaction Processing Concepts, Concurrency Control Techniques: Two-phase Locking, Timestamp Ordering, Multiversion, Validation, Multiple Granularity Locking.

Object Oriented and Object Relational Databases: Object Oriented Concepts, Object Oriented Data Model, Object Definition Language, Object Query Language, Object Relational Systems, SQL3, ORDBMS Design.

Distributed Databases: Distributed Database Concepts, Advantages and Disadvantages, Types of Distributed Database Systems, Data Fragmentation, Replication and Allocation Techniques for Distributed Database Design, Five Level Schema Architecture, Query Processing, Concurrency Control and Recovery in Distributed Databases.

Backup and Recovery: Types of Database Failures, Types of Database Recovery, Recovery Techniques: Deferred Update, Immediate Update, Shadow Paging, Checkpoints, Buffer Management.

Introduction to Data Warehousing and Data Mining: Introduction to OLAP, OLTP, Data Warehouse, Data Marts, Data Mining, Data Mining Process, Big Data.

Enterprise Database Products: Enterprise Database Products, Familiarity with IBM DB2 Universal Database, Oracle, Microsoft SQL Server, My SQL, their features.

COURSE OUTCOMES

After the completion of the course the student will be able:

CO1. To understand the basic concepts and design a database model using SQL and to apply Backup and Recovery techniques in case of failure in huge Databases and industry related applications.

CO2. To analyse various Query Processing and Evaluation Techniques and apply them for Query Optimization.

CO3. To understand the Transaction Processing and Concurrency Control techniques and apply these concepts in Object Oriented and Distributed databases.

CO4. To study the working of various real time database applications such as OLAP, OLTP, Data mining and Data ware housing and compare latest enterprise database products such as My Sql, Oracle etc.

List Of Recommended Books And Titles

1. Alexis Leon, Mathews Leon, Database Management Systems, Leon Press.
2. RamezElmasri, ShamkantNavathe, Fundamentals of Database Systems, Fifth Edition, Pearson
3. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, Database S ystem Concepts, Tata McGraw- Hill.
4. C.J. Date, An Introduction to Database Systems, Eighth Edition, Pearson Education
5. AtulKahate, Introduction to Database Systems

SOFTWARE ENGINEERING

BTCS-603

COURSE OBJECTIVES

1. This course gives the ability to understand the fundamental objective of Software Development Life Cycle (SDLC), which is the study of software evolution.

2. The course will provide a comprehensive knowledge about various aspects related to functionality and processing of software developments. An In-depth study about all phases of software development starting from feasibility to final implementation of the product which also include continuous update and maintains.

3. The focus of the course will be placed on understanding how the various phases of the SDLC interact and provides a base for the developments of a finished product catering to highest quality and service.

CONTENTS

Module1: Evolution and impact of Software engineering, software life cycle models: Waterfall, prototyping, Evolutionary, and Spiral models. Feasibility study, Functional and Non-functional requirements, Requirements gathering, Requirements analysis and specification.

Module2: Basic issues in software design, modularity, cohesion, coupling and layering, function-oriented software design: DFD and Structure chart, object modeling using UML, Object-oriented software development, user interface design. Coding standards and Code review techniques.

Module3: Fundamentals of testing, White-box, and black-box testing, Test coverage analysis and test case design techniques, mutation testing, Static and dynamic analysis, Software reliability metrics, reliability growth modeling.

Module4: Software project management, Project planning and control, cost estimation, project scheduling using PERT and GANTT charts, cost-time relations: Rayleigh-Norden results, quality management, ISO and SEI CMMI, PSP and Six Sigma. Computer aided software engineering, software maintenance, software reuse, Component-based software development.

Suggested Readings/ Books:

1. Roger Pressman, "Software Engineering: A Practitioners Approach,(6th Edition), McGraw Hill, 1997.
2. Sommerville,"Software Engineering, 7th edition", Adison Wesley, 1996.
3. Watts Humphrey," Managing software process", Pearson education, 2003.
4. James F. Peters and Witold Pedrycz, " Software Engineering – An Engineering Approach", Wiley.
5. Mouratidis and Giorgini. "Integrating Security and Software Engineering–Advances and Future", IGP. ISBN – 1-59904-148-0.
6. Pankaj Jalote, "An integrated approach to Software Engineering", Springer/Narosa.

COURSE OUTCOMES

After the completion of the course the student will be able to:

1. Understand, select and apply appropriate models and techniques for various phases of software development.
2. Apply design principles, coding guidelines and rigorous testing at various levels of software development.
3. Demonstrate knowledge and understanding of various standards for enhancing software quality and reliability.
4. Create, design, select and apply appropriate techniques, models and IT tools for software project management and being able to comprehend and write effective reports and documentation.

ETHICAL HACKING (ELECTIVE –I)

BTCS-903

COURSE OBJECTIVE

This course will provide the student with an overview of common methods and techniques used by attackers to penetrate a network and the methods which can be used to protect them.

CONTENTS

Introduction: Understanding the importance of security, Concept of ethical hacking and essential Terminologies-Threat, Attack, Vulnerabilities, Target of Evaluation, Exploit. Phases involved in hacking

Foot printing: Authoritative, Non -Auth reply by DNS, Introduction to foot printing, Understanding the information gathering methodology of the hackers, Tools used for the reconnaissance phase.

Scanning: Detecting live systems on the target network, Discovering services running /listening on target systems, Understanding port scanning techniques, Identifying TCP and UDP services running on the target network, Understanding active and passive fingerprinting.

System Hacking: Aspect of remote password guessing, Role of eavesdropping ,Various methods of password cracking, Keystroke Loggers, Understanding Sniffers ,Comprehending Active and Passive Sniffing, ARP Spoofing and Redirection, DNS and IP Sniffing, HTTPS Sniffing.

Hacking Wireless Networks: Introduction to 802.11, Role of WEP, Cracking WEP Keys, Sniffing Traffic, Securing Wireless Networks.

Cryptography: Understand the use of Cryptography over the Internet through PKI, RSA, MD-5, Secure Hash Algorithm and Secure Socket Layer.

COURSE OUTCOMES

After Completion of Course student will able to

CO1. Learn the fundamental principles and methods associated with the cyber security put into practice known as penetration testing or ethical hacking.

CO2. Become familiar with testing process including planning, reconnaissance, scanning, exploitation, post-exploitation and result reporting.

CO3. Develop a practical understanding of the current cyber security issues and the ways how the errors made by users, administrators, or programmers can lead to exploitable insecurities.

List Of Recommended Books And Titles

1. Network Security and Ethical Hacking, Rajat Khare , Luniver Press
2. Ethical Hacking, Thomas Mathew ,OSB Publisher
3. Hacking Exposed: Network Security Secrets & Solutions, Stuart McClure, Joel Scambray and George Kurtz, McGraw-Hill

HUMAN RESOURCE MANAGEMENT (OPEN ELECTIVE)

HU-251

COURSE OBJECTIVES

1. To familiarize the students with basic concepts, frameworks and models of human resource management (HRM).
2. To apply theory, concepts and models in practice using case studies.
3. To understand the role that HRM has to play in effective business administration.

CONTENTS:

Introduction: - Introduction to HRM and its definition, functions of Human Resource Management and its relation to other managerial functions. Nature, Scope and importance of HRM in industry, Role and position of Personnel function in the organization.

Procurement and Placement: - Need for HR Planning, Process of HR planning, Methods of recruitment, Psychological tests and interviewing; Meaning and Importance of Placement and Induction, Employment Exchanges(Compulsory Notification of Vacancies) Act 1959, The Contract Labour(Regulation and Abolition) Act 1970.

Training and Development: - Difference between training and development, Principles of Training, Methods and Types of Training, employee development; promotion-merit v/s seniority performance appraisal, career development and planning. Methods of Performance Appraisal.

Job analysis & design: - Job analysis, job description, and job specification, Job Evaluation, Job enlargement, Job Enrichment, job satisfaction and its importance, motivation, factors affecting motivation, introduction to various motivation theories, Industrial Democracy, Workers' participation, and quality of work life, quality circle.

The compensation Functions: - Basic concepts in wage administration, Company's wage policy, Issues in wage administration, Bonus and Incentives, Payment of wages Act-1936, Minimum wages act-1961(1948).

Integration: - Human relations and industrial relations, difference between human relation and industrial relation, Factors required for good human relation policy in industry, Employee Employer relationship, causes and effects of an industrial dispute, Employee Grievance And their redressal administration of discipline, communication in organization, absenteeism, labour turnover, changing faces of Indian workforce and their environment, Importance of Collective Bargaining, Role of Trade unions in maintaining cordial industrial relation.

Maintenance: - Fringe & retirement terminal benefits, administration of welfare amenities, meaning and importance of employee safety, accident and causes and their prevention, safety provisions under Factory Act 1948, Welfare of employee and its importance, Social security, Family pension scheme , Employee State Insurance (ESI) act 1948, workmen's gratuity act 1972, future challenges for human resources management

COURSE OUTCOMES

After the completion of the course the student will be able:

1. To understand basic functions of HRM, in order to be able to function/communicate effectively as individual, lead, team, groups and build human and industrial relations.

2. To plan and design effective strategies for recruitment, selection, training, and performance appraisal.

3. To set/study various standards and study existing laws to assess ethical, environmental, societal, health, safety and legal issues.

Recommended Text Books:

1. T.N.Chhabra- Human Resource Management (Dhanpat Rai & Co.)

Recommended Reference Books:

1. Lowin B. Flippo - Principles of personnel Management (Mc Graw-Hill)
2. R.C. Saxena - Labour Problems and social welfare (K.Math & Co.)
3. A Minappa and M. S. Saiyada - Personnel Management (Tata Mc. Graw-Hill)
4. C.B. Mamoria - Personnel Management (Himalaya Publishing House, Bombay)
5. T.N. Bhagotiwal - Economics of Labour and Industrial Relations (Sahitya Bhawan Agra)

RELATIONAL DATABASE MANAGEMENT SYSTEM -II LAB

BTCS-604

COURSE OBJECTIVES

1. Understand, appreciate and effectively explain the underlying concepts of database technologies.
2. Study the different issues relating to various aspects of a database design.

CONTENTS

1. Case studies on normalization
2. Study and usage of query optimization techniques
3. Study and usage of backup and recovery features of database management software
4. Server administration of any database management software
5. Study and usage of any object oriented or object relational database management software
6. Study and usage of open source data mining tool: Weka
7. Study of web databases
8. Development of a project by making use of tools studied above

COURSE OUTCOMES

After the completion of the course the student will be able:

CO1. To apply the normalization techniques for development of application software to realistic problems.

CO2. To present the issues and techniques relating to query optimization techniques, backup and recovery features, Server administration of any database management software.

CO3. To understand the concepts of Web Mining and Web Databases.

List Of Recommended Tools

Suggested Tools – MySQL, DB2, Oracle, SQL Server 2012, Postgre SQL, SQL lite

SOFTWARE ENGINEERING LAB

BTCS-606

COURSE OBJECTIVES

1. To understand the various paradigm for Software Development Life Cycle.

2. To design test cases and testing scenarios for software modules.

CONTENTS

1. Study and usage of OpenProj or similar software to draft a project plan
2. Study and usage of OpenProj or similar software to track the progress of a project
3. Preparation of Software Requirement Specification Document, Design Documents and Testing Phase related documents for some problems
4. Preparation of Software Configuration Management and Risk Management related documents.
5. Study and usage of any Design phase CASE tool
6. To perform unit testing and integration testing
7. To perform various white box and black box testing techniques
8. Testing of a web site.

COURSE OUTCOMES

Students will be able

CO1. To develop software utilizing functionalities of all phases of SDLC and various software paradigms.

CO2. To perform software testing using test cases.

SIMULATION AND MODELING LAB

BTCS 607

COURSE OBJECTIVES

1. To impart the fundamental knowledge on using various simulators for Engineering Simulation.
2. To impart knowledge on how these simulators are used in Industries by solving some real time problems using these tools.

CONTENTS

1. Programming in MATLAB: Introduction, Branching statements, loops, functions, additional data types, plots, arrays, inputs/outputs etc.

2. Introduction regarding usage of any Network Simulator.

3. Practical Implementation of Queuing Models using C/C++

COURSE OUTCOMES

Students will be able to

CO1. Appreciate the utility of the simulators in solving real time problems and day to day problems.

CO2. Develop models for Engineering Simulation.

CO3. Design and evaluate the performance of *queuing* systems.

Seventh/Eighth Semester

ARTIFICIAL INTELLIGENCE

BTCS 701

COURSE OBJECTIVES:

1. To familiarize the students with foundation, underlying theory and applications of artificial intelligence.
2. To enlighten the students with logic, knowledge and reasoning representation of various problem solving, planning paradigms and search strategies.
3. To familiarize the students with inductive learning and uncertainty principle and design algorithms for traditional AI problems.

CONTENTS:

Module1: Introduction- What is intelligence? Foundations of artificial intelligence (AI). History of AI; Problem Solving- Formulating problems, problem types, states and operators, state space, search strategies.

Module2: Informed Search Strategies- Best first search, A* algorithm, heuristic functions, Iterative deepening A*(IDA), small memory A*(SMA); Game playing - Perfect decision game, imperfect decision game, evaluation function, alpha-beta pruning.

Module3: Reasoning-Representation, Inference, Propositional Logic, predicate logic (first order logic), logical reasoning, forward chaining, backward chaining; AI languages and tools - Lisp, Prolog, CLIPS.

Module4: Planning- Basic representation of plans, partial order planning, planning in the blocks world, hierarchical planning, conditional planning, representation of resource constraints, measures, temporal constraints.

Module5: Uncertainty - Basic probability, Bayes rule, Belief networks, Default reasoning, Fuzzy sets and fuzzy logic; Decision making- Utility theory, utility functions, Decision theoretic expert systems.

Module 6: Inductive learning - decision trees, rule based learning, current-best-hypothesis search, least commitment search , neural networks, reinforcement learning, genetic algorithms; Other learning methods -neural networks, reinforcement learning, genetic algorithms.

Module7: Communication - Communication among agents, natural language processing, formal grammar, parsing, grammar.

COURSE OUTCOMES

After the completion of the course the student will be able:

CO1: Understand the fundamentals of Artificial Intelligence and its applications for problem solving.

CO2: Demonstrate working knowledge of AI search algorithms such as uninformed, informed and heuristic strategies.

CO3: Apply concepts of knowledge representation, planning and reasoning to real-world problems.

CO4: Understand the basics of inductive learning, uncertainty principles and natural language processing for machine learning

Suggested / Readings & Books

1. Stuart Russell and Peter Norvig. Artificial Intelligence – A Modern Approach, Pearson Education Press,
2001.
2. Kevin Knight, Elaine Rich, B. Nair, Artificial Intelligence, McGraw Hill, 2008.
3. George F. Luger, Artificial Intelligence, Pearson Education, 2001.
4. Nils J. Nilsson, Artificial Intelligence: A New Synthesis, Morgan Kauffman, 2002.

THEORY OF COMPUTATION

BTCS 702

COURSE OBJECTIVES

1. Introduce students to the mathematical foundations of computation including automata theory; the theory of formal languages and grammars; the notions of algorithm, decidability, complexity, and computability.
2. Enhance/develop students' ability to understand and conduct mathematical proofs for computation and algorithms.

CONTENTS:

Module1: Basics of Strings and Alphabets

Module2: Finite Automata – DFA, transition graphs, regular languages, non-deterministic FA, equivalence of DFA and NDFA

Module3: Regular grammars, regular expressions, equivalence between regular languages, properties of regular languages, pumping lemma.

Module4: Context Free Languages – Leftmost and rightmost derivation, parsing and ambiguity, ambiguity in grammar and languages, normal forms.

Module5: Pushdown Automata – NDPDA, DPDA, context free languages and PDA, comparison of deterministic and non-deterministic versions, closure properties, pumping lemma for CFL

Module6: Turing Machines, variations, halting problem, PCP

Module7: Chomsky Hierarchy, LR(k) Grammars, properties of LR(k) grammars, Decidability and Recursively Enumerable Languages.

COURSE OUTCOMES

After competition of course the student will able to:

1. Analyses and design finite automata, pushdown automata, Turing machines, formal languages, and grammars.
2. Be able to construct pushdown automata and the equivalent context free grammars.
3. Be able to prove the equivalence of languages described by pushdown automata and context free grammars.
4. Be familiar with thinking analytically and intuitively for problem-solving situations in related areas of theory in computer science.
5. Students will be able apply design and development principles in the construction of software systems of varying complexity.

SOFTWARE PROJECT MANAGEMENT (ELECTIVE –II)

BTCS-907

COURSE OBJECTIVES

1. To understand the fundamental principles of Software Project Management and have a good nowledge of responsibilities of a Project Manager.
2. To be familiar with different stages, methods and techniques of Software Project Planning.

3. To have a good knowledge of Project Scheduling, tracking, risk analysis, quality management and project cost estimation using different techniques.

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's 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, PERT and Monte Carlo Simulation techniques.

Monitoring And Control- Collecting Data, Visualizing Progress, Cost Monitoring, review techniques, project termination review, Earned Value analysis, Change Control, Software Configuration Management (SCM), Managing Contracts, Types Of Contracts, Stages In Contract Placement, Typical Terms of A Contract, Contract Management and Acceptance.

Quality Management and People Management- Introduction, Understanding Behaviour, 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, Organizational Structures, Stress, Health And Safety. ISO and CMMI models, Testing, and Software reliability, test automation, Overview of project management tools.

COURSE OUTCOMES

CO1. Understand and practice the process of Project Evaluation and Planning using different cost estimation and scheduling techniques thereby managing the risks.

CO2. Monitor and control the project thereby identifying project goals, constraints, performance criteria and resource requirements and produce an effective work plan by managing Contracts.

CO3. Understand Quality Management and People Management by interacting with team in a professional manner so as to ensure a collaborative work environment.

List Of Recommended Books And Titles

1. Bob Hughes, Mike Cotterell, "Software Project Management", Tata McGraw Hill. (2009)
2. Royce, "Software Project Management", Pearson Education. (2005).
3. Robert K. Wysocki, "Effective Software Project Management", Wiley.(2006)
4. Ian Sommerville, Software Engineering, Seventh Edition, Pearson Education.
5. R.S. Pressman, Software Engineering: A Practitioner's Approach, Sixth Edition, Tata McGraw-Hill.
6. Kassem, Software Engineering, Cengage Learning

CLOUD COMPUTING (ELECTIVE –III)

BTCS-912

COURSE OBJECTIVES

The objective of this course is to provide graduate students with the comprehensive and in-depth knowledge of Cloud Computing concepts, technologies, architecture and applications by introducing and researching state-of-the-art in Cloud Computing fundamental issues, technologies, applications and implementations. Another objective is to expose the students to frontier areas of Cloud Computing and information systems, while providing sufficient foundations to enable further study and research.

CONTENTS

Overview of cloud computing : What is a cloud, Definition of cloud , Definition of cloud ,characteristics of cloud ,Why use clouds, How clouds are changing , How clouds are changing , Driving factors towards cloud, Comparing grid with cloud and other computing systems, workload patterns for the cloud, "Big Data", IT as a service.

Cloud computing concepts: Concepts of cloud computing, Cloud computing leverages the Internet,Positioning cloud to a grid infrastructure, Elasticity and scalability, Virtualization, Characteristics of virtualization, Benefits of virtualization, Virtualization in cloud computing, Hypervisors, Multitenancy,Types of tenancy, Application programming interfaces (API), Billing and metering of services , Economies of scale, Management, tooling, and automation in cloud computing, Management: Desktops in the Cloud, Security**.**

Cloud service delivery: Cloud service , Cloud service model architectures, Infrastructure as a service (IaaS) architecture, Infrastructure as a service (IaaS) details, Platform as a service (PaaS) architecture, Platform as a service (PaaS) details, Platform as a service (PaaS) , Examples of PaaS software, Software as a service (SaaS) architecture, Software as a service (SaaS) details, Examples of SaaS applications, Trade-off in cost to install versus , Common cloud management platform reference architecture: Architecture overview diagram, Common cloud management platform.

Cloud deployment scenarios: Cloud deployment models, Public clouds, Hybrid clouds, Community, Virtual private clouds, Vertical and special purpose, Migration paths for cloud, Selection criteria for cloud deployment.

Security in cloud computing : Cloud security reference model, How security gets integrated , Cloud security , Understanding security risks, Principal security dangers to cloud computing, Virtualization and multitenancy, Internal security breaches, Data corruption or loss, User account and service hijacking, Steps to reduce cloud security breaches, Steps to reduce cloud security breaches, Reducing cloud security, Identity management: Detection and forensics, Identity management: Detection and Identity management, Benefits of identity, Encryption techniques, Encryption & Encrypting data , Symmetric key encryption, Asymmetric key encryption, Digital

signature, What is SSL? IBM Smart Cloud, Amazon Web Services, Google Cloud platform, Windows Azure platform, A comparison of Cloud Computing Platforms, Common building Blocks.

COURSE OUTCOMES

Upon successful completion of this course students should be able to:

CO1. Articulate the main concepts, key technologies, strengths, and limitations of cloud computing and the possible applications for state-of-the-art cloud computing

CO2. Identify the architecture and infrastructure of cloud computing, including SaaS, PaaS, IaaS, public cloud, private cloud, hybrid cloud, etc.

CO3. Compare, contrast, and evaluate the key trade-offs between multiple approaches to cloud system design, and Identify appropriate design choices when solving real-world cloud computing problems.

CO4. provide the appropriate cloud computing solutions and recommendations according to the applications used and attempt to generate new ideas and innovations in cloud computing

List Of Recommended Books And Titles

1. Raj Kumar Buyya, James Broberg, AndrezeiM.Goscinski, Cloud Computing: Principles and paradigms, 2011
2. Michael Miller, Cloud Computing, 2008.
3. Judith Hurwitz, Robin Bllor, Marcia Kaufman, Fern Halper, Cloud Computing for dummies, 2009.
4. Anthony T. Velte, Toby J. Velte and Robert Elsenpeter, Cloud Computing: A practical Approach, McGraw Hill, 2010.
5. Barrie Sosinsky, Cloud Computing Bible, Wiley, 2011.
6. BorkoFurht, Armando Escalante (Editors), Handbook of Cloud Computing, Springer, 2010.

ARTIFICIAL INTELLIGENCE LAB
BTCS-704

COURSE OBJECTIVES:

To introduce the basic concepts of artificial intelligence algorithms and logic programming.

CONTENTS:

1. Write A Program For DEPTH FIRST SEARCH

2. Write A Program For Best First Search

3. Write A Program to Generate the output for A* Algorithm.

4. Write a Lisp Program to solve Water Jug Problem Using Heuristic Function.
5. Write a Program To Show the Tic Tac Toe Game for 0 and X.
6. Write A Program For Expert System By Using Forward Chaining.
7. Write a program to implement tower of hanoi.
8. Write a program to implement a heuristic search procedure.
9. Write a program to implement a production system.
10. Write a program to implement search problems of 3 x 3 puzzle.

COURSE OUTCOMES:

1.To equip students with the knowledge and skills in logic programming;
2.To understand various Artificial intelligenece search algorithms (uninformed, informed, heuristic, genetic algorithms).
3.To explore the different paradigms in knowledge representation and reasoning;