DAV INSTITUTE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

Scheme and syllabus

Third Semester

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

Fourth Semester

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

Course Code	Course Name	Load Allocation			MarksDistribution		Total Marks	Credits
		L	T	P	Internal	External		
BTIT501	System Analysis and Design	3	1	-	40	60	100	4
BTIT502	Programming in Java	3	1	-	40	60	100	4
BTIT503	Database Management Systems	3	1	-	40	60	100	4
BTCS501	Computer Networks –II	3	1	-	40	60	100	4
BTIT504	Cyber Laws and IPR	3	-	-	40	60	100	3
BTIT505	Database Management Systems Lab	-	-	4	30	20	50	2
BTIT506	Programming in Java Lab	-	-	4	30	20	50	2
BTCS507	Computer Networks –II Lab	-	-	2	30	20	50	1
BTIT507	Industrial Training	-	-	-	60	40	100	1
Total		15	4	10	350	400	750	25

Sixth Semester

Course Code	Course Name	Load Allocation			Marks Distribution		Total Marks	Credits
		L	T	P	Internal	External		
BTIT601	Network Programming	3	1	-	40	60	100	4
BTIT602	Information Security and Risk Management	3	-	-	40	60	100	3
BTIT603	Web Technologies	3	-	-	40	60	100	3
BTCS603	Software Engineering	3	1	-	40	60	100	4
BTXXX	Elective –I	3	1	-	40	60	100	4
BT***	Open Elective	3	1	-	40	60	100	4
BTIT604	Network Programming Lab	-	-	2	30	20	50	1
BTCS606	Software Engineering Lab	-	-	2	30	20	50	1
BTIT605	Web Technologies Lab	-	-	4	30	20	50	2
General Fitness					100	-	100	
Total		18	03	08	430	420	850	26

Seventh Semester

Course Code	Course Name	Load Allocation			Marks Distribution		Total Marks	Credits
		L	T	P	Internal	External		
BTIT701	Building Enterprise Applications	3	-	-	40	60	100	3
BTIT702	Software Project Management	3	1	-	40	60	100	4
BTIT 703	Project	-	-	12	150	150	300	12
BTYYY	Elective –II	3	1	-	40	60	100	4
BTZZZ	Elective –III	3	1	-	40	60	100	4
BTIT704	Building Enterprise Applications Lab	-	-	2	30	20	50	1
General Fitness					100	-	100	
Total		15	03	04	370	380	750	22

THIRD SEMESTER

BTCS-301 COMPUTER ARCHITECTURE

COURSE OBJECTIVES

This course offers a good understanding of the various functional units of a computer system and prepares the student to be in a position to design a basic computer system.

SYLLABUS

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. [5]

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. [6]

3. Design of Control Unit: Control memory, design of control unit – microprogrammed, hardwired, and their comparative study. [3]

4. Central Processing Unit: General Register Organisation, Stack Organisation, Instruction formats, Addressing Modes, Data transfer and manipulations, Program control, RISC and CISC architecture. [6]

5. Input-Output Organisation: Peripheral devices, I/O Interface, asynchronous data transfer, modes of transfer, priority interrupt, DMA, I/O processor, serial communication. [5]

6. Memory Organisation: Memory hierarchy, main memory, auxiliary memory, associative memory, cache memory, virtual memory, memory management hardware. [6]

7. Advanced concepts of Computer Architecture: Concept of pipeline, Arithmetic pipeline, Instruction , vector processors and array processors. Introduction to parallel processing, Interprocessor communication & synchronization [5]

COURSE OUTCOMES

CO1 Ability to understand how computer hardware has evolved to meet the needs of multiprocessing systems, Instruction Set Architecture: Instruction format, types, various addressing modes, the basic components and design of the CPU: the ALU and control unit.

CO2 Understand the memory organization: SRAM, DRAM, concepts on cache memory, Memory Interleaving, Associative memory, Virtual memory organization.

CO3 Ability to understand the parallelism both in terms of a single processor and multiple processors.

CO4 Understand the I/O Organization: Basics of I/O, Memory-mapped I/O & I/O mapped I/O, types of I/O transfer: Program controlled I/O, Interrupt-driven I/O, DMA

BTCS 304 DATA STRUCTURES

OBJECTIVES:

This course should provide the students with

1. A fairly good concept of the fundamentals of different types of data structures and the various operations that can be performed on them.
2. Algorithm for solving problems like sorting, searching, insertion & deletion of data etc. related to data structures.
3. Ability to choose an appropriate data structure for a particular problem.

SYLLABUS:

1. Dynamic Memory Management: Understanding pointers, usage of pointers, arithmetic on pointers, memory allocation, memory management functions and operators, debugging pointers - dangling pointers, memory leaks, etc.
2. 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.
3. Arrays: Linear and multi-dimensional arrays and their representation, operations on arrays, sparse matrices and their storage.
4. Linked List: Linear linked list, operations on linear linked list, doubly linked list, operations on doubly linked list, application of linked lists.
5. 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.
6. 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.
7. 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.
8. Heaps: Representing a heap in memory, operations on heaps, application of heap in implementing priority queue and heap sort algorithm.
9. 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.

10. 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.
11. 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.

Suggested Readings/Books:

1. Sartaj Sahni, Data Structures, Algorithms and Applications in C++, Tata McGraw Hill.
2. Tenenbaum, Augenstein, & Lan sam, 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. R. S. Salaria, Test Your Skills in Data Structures.
7. Malik, Data Structures using C++, Cengage Learning.

COURSE OUTCOMES:

After completing this course students will be able to:

CO 1. Understand the concept of Dynamic memory management, data types, algorithms, Big O notation.
CO 2. Understand basic data structures such as arrays, linked lists, stacks and queues.
CO 3. Describe the hash function and concepts of collision and its resolution methods
CO 4. Solve problem involving graphs, trees and heaps
CO 5. Apply Algorithm for solving problems like sorting, searching, insertion and deletion of data

BTCS306 DATA STRUCTURES LAB

OBJECTIVES:

The objectives of this course are:

1. To understand the programming constructs used for implementing different data structures
2. To implement various data structures using a programming language
3. To implement the various operations like searching, sorting.

SYLLABUS:

1. Write a menu driven program to implement following operations on linear array:
   i. Traversing an array
   ii. Insert a new element at end as well as at a given position

iii. Delete an element from a given whose value is given or whose position is given
iv. Searching an element using linear search
v. Searching an element using binary search

2. Write a menu driven program to implement following operations on linked lists:
   i. Insert a new element
   ii. Delete an existing element
   iii. Search an element
   iv. Display all the elements
3. Write a program to demonstrate the use of stack (implemented using linear array) in converting arithmetic expression from infix notation to postfix notation.
4. Program to demonstrate the use of stack (implemented using linear linked lists) in evaluating arithmetic expression in postfix notation.
5. Program to demonstrate the implementation of various operations on a linear queue represented using a linear array
6. Program to demonstrate the implementation of various operations on a circular queue represented using a linear array.
7. Program to demonstrate 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. Write a menu driven program to show graph traversals.
10. Write a menu driven program to implement various sorting algorithms

COURSE OUTCOMES:

After completing this course students will be able to:

CO 1. Improve practical skills in implementing data structure algorithms.
CO 2. Design menu-driven interface for better human computer interaction.
CO 3. Programs to demonstrate the implementation of various operations on data structure
CO 4. Utilize data structure algorithms in a better way to solve responsibilities relevant to other professional engineering practices

OBJECT ORIENTED PROGRAMMING USING C++ (BTCS305)

COURSE OBJECTIVES:

To understand the basic concepts of object oriented programming languages and to learn the techniques of software development in C++.

SYLLABUS:

1. 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.

2. 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.

3. 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.

4. 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.

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

6. 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.

7. 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.

8. Virtual functions & Polymorphism: Concept of binding - early binding and late binding, virtual functions, pure virtual functions, abstract clasess, virtual destructors.

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

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

11. 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 studying this course, the students will be able to

CO1. Gain knowledge on basics of object oriented programming.
CO2. Understand input/output streams used in C++.
CO3. Understand the concept of classes and objects.
CO4. Gain knowledge on different concepts related to pointers.
CO5. Understand constructor, destructor and operator overloading functions.
CO6. Gain knowledge on inheritance and polymorphism concepts.
C07. Understand template, file handling, exception handling concepts.

BTCS 309 Object Oriented Programming Using C++ Lab

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++.

SYLLABUS

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 studying this course the students will be able to:

1. Develop solutions for a range of problems using objects and classes.
2. Programs to demonstrate the implementation of constructors, destructors and operator overloading.
3. Apply fundamental algorithmic problems including type casting, inheritance, and polymorphism.
4. Understand generic programming, templates, file handling.

BTCS303 DIGITAL CIRCUITS & LOGIC DESIGN

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.

SYLLABUS

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.

BTCS-402 ENGINEERING MATHEMATICS-III

COURSE OBJECTIVES

1. To enhance the multi-dimensional skills namely problem defining, analysis, logic development, path making for solution and selecting appropriate technique of solution.
2. Content Objectives: Students should learn fundamental mathematical concepts and how to apply them.
3. Skill Objectives: Students should learn critical thinking, modeling/problem solving and effective uses of technology
4. Communication Objectives: Students should learn how to read mathematics and use it to communicate knowledge.

SYLLABUS

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

Part- A

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

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

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

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

Part- B

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

.

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

7. Probability distribution: Binomial, Poisson and Normal distribution.

8. 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

After successful completion of the course the students are able to

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 and to apply their usage in solving differential equations.

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.

FOURTH SEMESTER

(BTCS-405) SYSTEM PROGRAMMING

COURSE OBJECTIVES: This course provides knowledge to design various System Programs

SYLLABUS:

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

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

Unit 3. Macro processors: Description macro expansion. macros, macro expansion, conditional and Recursive.

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

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

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

Unit 7. Debuggers: Description of various debugging techniques.

COURSE OUTCOMES:

After the completion of the course the student:

1. will be able to understand the basics of system programs like editors, compilers, assemblers, macro-processors, compilers, linkers, loader, debuggers.
2. will have sufficient knowledge of various phases of compiler and will be able to compare its working with assembler.
3. will understand how linker and loader create an executive program from an object module created by assembler and compiler.
4. will know various editors and debugging techniques.

BTCS 409 SYSTEM PROGRAMMING LAB

SYLLABUS:

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.

(BTCS403) COMPUTER NETWORKS–I

COURSE OBJECTIVES

1. This course gives the ability to understand the fundamental of Computers Networks and basic functions of OSI Model.
2. The course will provide a comprehensive knowledge about various aspects related to various layers of OSI Model, their functions and role in data transmission. The course is aimed at providing a detail conceptual knowledge about data communication.
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.

SYLLABUS

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, ISOOSI 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: [6] Design issues, Framing, Error detection and correction codes: checksum, CRC, hamming code, Data link protocols for noisy and noiseless channels, Sliding Window Protocols: Stop & ARQ, Selective repeat ARQ, Data link protocols: HDLC and PPP.
4. Medium Access Sub-Layer: Wait ARQ, Go-back-N [6] 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: [6] 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: [6] 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: [3] World Wide Web (WWW), Domain Name System (DNS), E-mail, File Transfer Protocol (FTP), Introduction to Network security

COURSE OUTCOMES

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

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 applications.

CO3. Study all seven layers related to OSI model with complete structure, functions and their role in data communication.

CO4. Use of network devices at various levels of data transmission.

CO5. A comprehensive analysis of security aspect related to computer networks.

(BTCS 407) COMPUTER NETWORKS-I LAB

COURSE OBJECTIVES

1. Familiarisation of basic fundamentals of data communication and networks.

2. To establish various topologies, implement client-server networks.

3. Knowledge of networking devices and tools.

SYLLABUS

1.To study various topologies for establishing computer networks.

2. To learn the usage of various basic tools (crimping, krone etc.) used in establishing a LAN.

3. To familiarize with switch and hub used in networks

4. To learn the usage of connecters and cables (cabling standards) used in networks

5. To make certain copper and fiber patch cords using different standards.

6. To familiarize with routers & bridges

7. Use commands like ping, ipconfig for trouble shooting network related problems.

8. Develop a program to compute the Hamming Distance between any two code words.

9. Develop a program to compute checksum for an ‘m’ bit frame using a generator polynomial.

COURSE OUTCOMES

After attending Lab, students will

CO1. Implementation of basic network topologies and their application.

CO2. Able to understand the functional concept of Switches, hubs or routers.

CO3. Configure basic commands on computer networks.

BTCS401 OPERATING SYSTEMS

COURSE OBJECTIVES:

1. This course provides a comprehensive introduction to understand the underlying principles, techniques and approaches which constitute a coherent body of knowledge in operating systems.
2. The course will consider inherent functionality and processing of program execution. Good conceptions on all the subjects like processes, inter-process communication, semaphore, message passing, classical IPC problems, scheduling, memory management, file systems, security and protection mechanism, I/O hardware and software, deadlocks, etc. should be provided.
3. The emphasis of the course will be placed on understanding how the various elements that underlie the operating system interact and provides services for the execution of application software.

SYLLABUS:

Unit1. Introduction to Operating system, Role of Operating System as resource manager, function of kernel and shell, operating system structures, views of an operating system.

Unit2. Process management: CPU scheduling, Scheduling Algorithms, PCB, Process synchronization, Deadlocks, Prevention, Detection and Recovery.

Unit3. 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

Unit4. Device Management: I/O system and secondary storage structure, Device management policies, Role of I/O traffic controller, scheduler

Unit5. File Management: File System Architecture, Layered Architecture, Physical and Logical File Systems, Protection and Security.

Unit6. Brief study to multiprocessor and distributed operating systems.

COURSE OUTCOMES :

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

CO1. Identify the functions, structure and design issues associated with operating systems.

CO2. Familiarize with different concepts of process management like inter-process communication, semaphore, message passing, classical IPC problems, CPU scheduling, deadlock detection and prevention.

CO3. Understand and analyze the theory and implementation of physical and virtual memory, memory management policies, I/O system and secondary storage structure.

CO4. Implement the different CPU scheduling, page replacement and device algorithms in C++.

CO5. Install the various types of operating systems including UNIX, Windows.

CO6. Contrast and compare the various file systems and their corresponding protection and security mechanisms.

BTCS 406 OPERATING SYSTEM LAB

COURSE OBJRCTIVES

1. To familiarize the students with the Operating System.
2. To demonstrate the process, memory, file and directory management issues under the UNIX/ LINUX operating system
3. To introduce LINUX basic commands

4. To make students how to make simple programs in LINUX and administrative task of LINUX

SYLLABUS

1. Installation Process of various operating systems

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

3. Commands for files & directories: cd, ls, cp, md, rm, mkdir, rmdir. Creating and viewing files using cat. File comparisons.linux, connecting processesManual help. Background Disk related commands: checking disk free spaces. Processes in with pipes, background processing, managing multiple processes. 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

After completing this course the students will be able to:

1. Demonstrate the installation process of various operating systems.
2. Implement virtualization by installing Virtual Machine software.
3. Apply Unix/Linux operating system commands.
4. Understand different Unix/Linux shell scripts and execute various shell programs.

BTCS402 DISCRETE STRUCTURES

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.

SYLLABUS

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, Karnaugh-map).

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: Directed & Indirected, Eulerian Chains & cycle, Hamiltonian chains & cycles, Trees, Chromatic number connectivity, Graph Coloring, Connected Graphs, Isomorphism & Homomorphism.

COURSE OUTCOMES

CO1. Perform operations on various discrete structures such as sets, functions, relations, and sequences.
CO2. Ability to solve problems using Counting techniques, Permutation and Combination, Recursion and generating functions.
CO3. Apply algorithms and use of graphs and trees as tools to visualize and simplify Problems.
CO4. Apply algorithms and use of graphs and trees as tools to visualize and simplify Problems.
CO5. Use of K-Maps and Truth Tables to construct and verify correctness of a Boolean expression.
CO6. Understand the various properties of algebraic systems like Rings, Monoids and Groups.

BTCS404 MICROPROCESSORS AND ASSEMBLY LANGUAGE PROGRAMMING

COURSE OBJECTIVES

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

SYLLABUS

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.

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

COURSE OUTCOMES:

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

CO1. Describe the general architecture of a microcomputer system and architecture organization of 8085 and understand the difference between 8085& 8086 Microprocessor and advanced microprocessors.

CO2 Understand and classify the instruction set of 8085 microprocessor and distinguish theuse of different instructions and apply it in assembly language programming.

CO3 Understand and realize the Interfacing of various I/O devices with 8085microprocessor

MICROPROCESSOR & ASSEMBLY LANGUAGE PROGRAMMING LAB.

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.

SYLLABUS

BTCS408 Microprocessor and Assembly Language Programming Lab
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.

FIFTH SEMESTER

B.TECH IT-501 SYSTEM ANALYSIS & DESIGN

COURSE OBJECTIVES:

1. The course has been designed to provide a solid foundation of systems principles
2. An understanding of how business function.
3. While heightening students to the issues analysts face daily.

SYLLABUS

1. Introduction: System definition and concepts: Characteristics and types of system, Manual and automated Systems, Real-life Business sub-systems: Production, Marketing, Personal, Material, Finance. Systems models types of models: Systems environment and boundaries, Real-time and distributed systems, Basic principles of successful systems

2. Systems analyst: Role and need of systems analyst, Qualifications and responsibilities, Systems Analyst as an agent of change

3. System Development cycle: Introduction to systems development life cycle (SDLC) : Various phases of development :Analysis, Design, Development, Implementation, Maintenance Systems documentation considerations: Principles of systems documentation, Types of documentation and their importance, Enforcing documentation discipline in an organization .

4. System Planning: Data and fact gathering techniques: Interviews, Group communication, Presentations, Site visits. Feasibility study and its importance, Types of feasibility reports System Selection plan and proposal Prototyping .Cost-Benefit and analysis: Tools and techniques

5. Systems Design and modeling: Process modeling, Logical and physical design, Design representation, Systems flowcharts and structured charts , Data flow diagrams , Common diagramming conventions and guidelines using DFD and ERD diagrams. Data Modeling and systems analysis , Designing the internals: Program and Process design ,Designing Distributed Systems **6. Input and Output:** Classification of forms: Input/output forms design, User-interface design, Graphical interfaces

7. Modular and structured design: Module specifications, Module coupling and cohesion, Top-down and bottom-up design.

8. System Implementation and Maintenance: Planning considerations, Conversion methods, producers and controls, System acceptance Criteria, System evaluation and performance, Testing and validation, Systems qualify Control and assurance, Maintenance activities and issues.

9. System Audit and Security: Computer system as an expensive resource: Data and Strong media Procedures and norms for utilization of computer equipment, Audit of computer system usage, Audit trail Types of threats to computer system and control measures: Threat to computer system and control measures, Disaster recovery and contingency planning

10. Object Oriented Analysis and design: Introduction to Object Oriented Analysis and design life cycle, object modeling: Class Diagrams, Dynamic modeling: state diagram, Dynamic modeling: sequence diagramming.

11. Case study of the following systems:

(I) Inventory Control

(II) Railway Reservation System

(III) University Management System

(IV) Hospital management System

TEXTS BOOKS

1. System analysis and design - Elias M.Awad.

REFERENCES

1. System analysis and design –Perry Edwards

2. Analysis and design of information systems – James A.Senn

COURSE OUTCOMES

CO1. Describe the concepts of systems analysis and information systems development.

CO2. Develop and analyze the systems requirements documentation

CO3. Analyze the system modeling & design

CO4. Identify system implementation, testing techniques & system security.

CO5. Describes the system audit trails.

CYBER LAWS AND IPR (BTIT 504)

COURSE OBJECTIVES

1. This course explains the cyber laws that are in place to keep cybercrimes in check. In addition to cyber laws, it elaborates various IT Security measures that can be used to protect sensitive data against potential cyber threats.

2. This course encompassing all relevant IP legislations in India with a view to understand and adjust with changing needs of the society because creative work is useful to society and law relating to innovation/creativity.

3. The course is designed with a view to create IPR consciousness; and familiarize the learners about the documentation and administrative procedures relating to IPR in India.

SYLLABUS:

Basics of Computer & Internet Technology:

Internet, ISP & domain name; Network Security; Encryption Techniques and Algorithms; Digital Signatures

Introduction to Cyber World:

Introduction to Cyberspace and Cyber Law; Different Components of cyber Laws; Cyber Law and Netizens.

E-Commerce:

Introduction to E-Commerce; Different E-Commerce Models; E-Commerce Trends and Prospects; E-Commerce and Taxation; Legal Aspects of E-Commerce.

Intellectual Property Rights:

IPR Regime in the Digital Society; Copyright and Patents; International Treaties and Conventions; Business Software Patents; Domain Name Disputes and Resolution.

IT ACT 2000:

Aims and Objectives; Overview of the Act; Jurisdiction; Role of Certifying Authority; Regulators under IT Act; Cyber Crimes-Offences and Contraventions; Grey Areas of IT Act. Project Work: Candidates will be required to work on a project. At the end of the course students will make a presentation and submit the project report.

COURSE OUTCOMES

After the completion of the course the student:

CO1. Will be able to understand the basics of Internet, ISP, domain name and various Network Security techniques.

CO2. Will have sufficient knowledge of Cyberspace, Cyber Law, E-Commerce and Taxation

CO3. Will understand various IPR Regime in the Digital Society, Copyright and Patents, International Treaties and Conventions, Business Software Patents, Domain Name Disputes and Resolution.

CO4. Will know the Overview of the Act, Jurisdiction, Role of Certifying Authority, Regulators under IT Act, Cyber Crimes-Offences and Contraventions, Grey Areas of IT Act.

BTIT 502 PROGRAMMING IN JAVA

COURSE OBJECTIVES: This course will provide the knowledge of java and prepare students to be in a position to write object oriented programs in Java.

SYLLABUS

Overview of Java: Object oriented programming, Two paradigms, abstraction, the three OOP principles, Java class libraries

Data types, Variables and Arrays: Integers, floating-point types, characters, Boolean , Iterates, Variables, Data types and casting, automatic type promotion in expressions, arrays.

Operators and Control Statements: Arithmetic operators, bit wise operators, relational operators, Boolean logical operators, the ? operator, operator precedence, Java’s selection statements, iteration statements, jump statements.

Introduction to Classes: Class fundamentals, declaring object reference variable, Introducing methods, constructors, this keyword, garbage collection, the finalize() method

Methods and Classes: Overloading methods, using objects as parameters, recursion.

Inheritance: Inheritance basics, using super, method overriding, dynamic method dispatch, using abstract classes, using final with inheritance, Package and Interfaces, package access protection, importing packages

Exception Handling: Exception handling fundamentals, Exception types, Uncaught exceptions, Using try and catch, multiple catch clauses, nested try statements, throw, finally, Java’s built-in exceptions, creating your own exception sub classes, using exceptions

Multithreaded Programming: The Java thread model, the main thread, creating thread, creating multiple threads, using isalive() and join(), Thread priorities, synchronization, Inter thread communications, suspending resuming and stopping threads.

String Handling: The string constructors, string length, special string operations, character extraction, string comparison, searching string, modifying string, data conversion, changing the case of characters, string buffer.

I/O and Applets: I/O Basics, Reading Console Input, Writing Console Output, Reading and Writing files, Applet Fundamentals, Applet Architecture, The HTML Applet Tag, Passing parameters to Applets.

Networking: Networking basics, Java and the Net, TCP/IP Client Sockets URL, URL Connection, TCP/IP Server Sockets, Database connectivity.

COURSE OUTCOMES

CO1: Gain knowledge about basic Java language syntax and semantics to write basic Java programs.

CO2: Understand the concept of Classes classes, object creation, invoking methods etc and exception handling mechanisms, reusability of classes etc.

CO3:The student will demonstrate an understanding of multithreading.

CO4: Understand the basic principles of creating Java applications with graphical user interface (GUI) and achieving networking capabilities.

(BTIT503) DATABASE MANAGEMENT SYSTEMS

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 Data Warehouse, Data Mining and Knowledge Discovery.
4. To train the students with structure query language and its use in data base management.

SYLLABUS

Introduction - Database Systems versus File Systems, View of Data, Data Models, database languages, Database Users and Administrators. Transaction Management, Decision Support Systems, Components of a Database management System. Distributed Processing and Client-Server Architecture.

Entity-Relationship Model – Basic Concepts, Constraints, Keys, Design Issues, E-R Diagrams. Relational Model - Structures of relational databases, Integrity Constraints, Logical database Design, Tables, Views, Data Dictionary. Relational Algebra, Relational Calculus. SQL – Basic Structures, Query Handling, Embedded SQL, Open Database Connectivity (ODBC), Java Database Connectivity (JDBC), Triggers, Security and Authorization. Query By Example (QBE), User Interfaces and Tools, Forms and Graphical User Interfaces. Report Generators. Overview of Relational Query Optimization.

Relational Database Design - Functional Dependencies, Multi-valued Dependencies, Normal Forms, Decomposition into Normalized Database Design – File Structures. Object-Relational Databases Complex Data types, Object-Relational Features in SQL:1999. Relations, Physical – Nested Relations, Internet Databases - World Wide Web, Client Side Scripting and Applets, Web Servers and Sessions, Services, Server Side Scripting. XML – Structure of XML Data, XML Document Schema, XQuery, Storage of XML Data, XML Applications.

Advanced Topics - Fundamental Concepts of Transaction Management, Concurrency Control, Recovery Systems, Data Analysis and OLAP. Introduction to Data Mining, Data Farming, Data Warehousing, Spatial and Geographic Databases, Temporal databases and Multimedia Databases. NoSQL databases

COURSE OUTCOMES

Students will be able to:
CO1. Understand the concepts of Database and DBMS with its applications.
CO2. Conceptualize and design data bases model using ER diagram for real world applications.

CO3. Familiarization of Relational data bases and Normalisation.
CO4. Learning of advance concepts of Transaction Management, Concurrency Control and Recovery.

(BTIT505) DATABASE MANAGEMENT SYSTEMS LAB

COURSE OBJECTIVES

1. To train the students with structure query language and its use in data base management.
2. To provide a hand on experience with live projects based on data bases.

SYLLABUS

1.Introduction to SQL and installation of a database product

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, 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.

9. Working with XML databases

10. Introduction to a data mining tool

COURSE OUTCOMES

After attending Lab sessions, students will

CO1: Able to create, update and implement data base with the help of structured query language on platforms like ORACLE and MYSQL.

CO2: Prepare for industry certification viz. OCA, OCP.

BTCS 501 COMPUTER NETWORKS-II

COURSE OBJECTIVES:

1. To familiarize the students of IT branch in engineering with computer networks, wireless

networks, adhoc networks, cellular networks

2. To familiarize the students with the need of IPV6 over IPv4 and understand the concepts of Security at the network layer.

SYLLABUS

Unit 1 Network Security: Fundamentals of network security, Basics of IPv6, IPsec: overview of IPsec, IP and IPv6, Authentication header (AH), Encapsulating Security Payload (ESP).

Unit 2 Internet Key Exchange (IKE): History, Photuris, Simple Key-management for Internet protocols (SKIP), IKE phases, IKE encoding.

Unit 3 Adhoc networks: Features, advantages and applications, Adhoc versus Cellular networks, Network architecture, Protocols: MAC protocols, Routing protocols, Technologies.

Unit 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.

Unit 5 3G wireless networks: wireless local loop (WLL), Local Multipoint Distribution System (LMDS), Wireless local Area Networks (WLANs), Bluetooth and Personal Area Networks.

Unit 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:

After studying the course, the student must be able to-

CO1 Explain the concepts of IPv6 and the need for implementing it.

CO2 Describe the evolution and History of Wireless Technology

CO3 Explain the function of Mobile station roaming number

CO4 Describe the basic operation of GSM GPRS

CO5 Differentiate between Wireless LAN and Wired LAN

CO6 Demonstrate use of various private and public key encryption techniques used in modern cryptosystems (IKE)

CO7 Understand the need, features and applications of Mobile Adhoc Networks

CO8 Describe various Routing and MAC protocols used in Mobile Adhoc Networks.

SIXTH SEMESTER

BTIT601 NETWORK PROGRAMMING

COURSE OBJECTIVES:

1. The main objective of this course is to make students able to understand the concepts of inter process communication.
2. The emphasis of this course will be placed on implementation of various inter process communication tools in Linux environment.

SYLLABUS:

Unit 1

OSI model, client server model, TCP/IP protocols, Introduction to Unix; Process, groups, job control and non-job control shells, reliable and unreliable signals, shell Programming.

Unit 2

Inter process communication in Unix, pipes, half duplex and full duplex pipes, FIFOs, properties of pipes and FIFOs, POSIX message queues, system V message queues, semaphores, shared memory, mmap function and its use, RPC, authentication, timeout and retransmission, call semantics, XDR.

Unit 3

Communication Protocol – Introduction, TCP, IP, XNS, SNA, NetBIOS, OSI protocols, comparisons.

Unit 4

Introduction to Berkeley sockets, socket addressing, TCP and UDP socket functions, sockets and Unix signals, socket implementation, client and server examples for TCP and UDP and their behavior under abnormal conditions.

Unit 5

Socket options, IPv4, IPv6, TCP, I/O multiplexing, Unix I/O models, select and poll functions

Unit 6

System V Transport Layer, interface – Introduction Transport End Point address, TLI.

COURSE OUTCOMES:

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

CO1. Implement various Inter Process Communication techniques in Linux/Unix environment.

CO2. Explore various Network APIs available in Linux/Unix environment.

CO3. Comprehend and compare various communication protocols

CO4. Understand socket options at different layers.

INFORMATION SECURITY AND RISK MANAGEMENT (BTIT-602)

COURSE OBJECTIVES:

1. To familiarize the students with good understanding of fundamentals of computer security and information security.
2. To familiarize the students with risk management concepts.
3. To familiarize the students with various encryption and decryption algorithms used in information security.

SYLLABUS:

1. **Essentials of Information Security, Security Threats –** Intruders, Viruses, Worms, and other
Threats, Vulnerabilities, Cyber Crime and Hacker, Security Assessment, Analysis and Assurance, Role of Cryptography, The Data encryption Standard (DES), Analyzing and Strengthening of DES, Introduction to Advance Encryption Standard (AES)

2. **Concept and Characteristics of Public Key Encryption system**, Introduction to Merkle-Hellman Knapsacks, Rivets – Shamir-Adlman (RSA) Encryption. Hash Algorithms, Message Digest Algorithms such as MD4 and MD5, Secure Hash Algorithms such as SH1 and SHA2.Digital Signature Standard.

3. **Network Security Issues such as Impersonation**, Message Confidentiality, Message Integrity,
Code Integrity, Denial of Service, Firewalls, DMZs, Virtual Private Networks. Web Security, Email Security, Pretty Good Privacy

4. **Risk Management and Security planning –**Risk management Process Overview and Life Cycle Activities, Information Security Life Cycle, Risk Analysis, Cost Benefit Analysis, Risk Assessment Process, Methodology, Threat assessment, Modes of risk analysis – Effective Risk analysis, Risk Mitigation, Qualitative Risk Analysis, Value Analysis, Case Study of IT Organization

COURSE OUTCOMES:

After the completion of the course the student:

CO1. Will have to understand why security and its management are important for any modern organization.

CO2. Will be able to study and write various algorithms used for encryption and decryption in cryptosystem.

CO3. Will understand how an information security management system should be planned, documented, implemented and improved, according to the information security management.

CO4. Will have to understand the concept of risk, resolving risks in information technology.

(BTIT 603) WEB TECHNOLOGIES

COURSE OBJECTIVES:

On completion of this course, a student will be familiar with client server architecture and able to develop a web application using java and other client side technologies.

SYLLABUS

Unit-1

Internet and World Wide Web: Introduction, Internet Addressing, ISP, types of Internet Connections, Introduction to WWW, WEB Browsers, WEB Servers, URLs, HTTP, WEB Applications, Tools for web site creation.

Unit-2

HTML5: Introduction to HTML5, Lists, adding graphics to HTML5 page, creating tables, linking documents, forms, frames, Cascading Style sheets.

Unit-3

Java Script: Introduction, programming constructs: variables, operators and expressions, conditional checking, functions and dialog boxes, JavaScript DOM, creating forms, introduction to Cookies, Jquery.

Unit - 4

AJAX: Introduction, HTTP Request, XMLHttpRequest, AJAX Server Script.

Unit-5

PHP: Introduction, syntax, statements, operators, PHP and MySQL, PHP and AJAX.

Unit-6

Introduction to ASP.net, J2EE, POJO, Java servlets and JSP.

COURSE OUTCOMES

After completing this course students will be able to:

CO1. Students will have clear understanding of hierarchy of objects in HTML and XML.

CO2. Students will be able to validate user input using JavaScript.

CO3. Students will have sufficient knowledge of advantages and applications of different types of CSS.

CO4. Students will have knowledge of basics of ASP, AJAX and various other aspects of web technologies.

BTIT605 WEB TECHNOLOGIES LAB

COURSE OBJECTIVES: On completion of this course, Students will gain the skills and project-based experience needed for entry into web application and development careers.

SYLLABUS:

1. Creation of Web pages using HTML, DHTML.
2. Creation of Web pages using JavaScript.
3. Creation of Web pages using AJAX.
4. Creating web pages using PHP.
5. Creating Web pages using ASP.

COURSE OUTCOMES

After completing this course the Students will be able to:

CO1.Develop interactive and innovative web pages using various platforms like HTML and HTML5, JavaScript, PHP, AJAX, JQUERYetc.

(BTCS 603) SOFTWARE ENGINEERING

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.

SYLLABUS

Module1: Evolution and impact of Software engineering, software life cycle models: Waterfall, prototyping, Evolutionary, and Spiral models. Feasibility study, Functional and Non-functionalrequirements, 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.

COURSE OUTCOMES

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

CO1. Understand the complete Software life cycle models for development of various functional Software's.
CO2. Know about the working of each phase for software development form feasibility study to final deployment of the software application**.**
CO3. Analyze the various software performance metric like effort of development and time for development of a software product which are very critical for software delivery.
CO4. Focus on the fundamental steps on software development which include rigorous testing at various levels of the product before final installation.
CO5. Ensure to compile the final product with highest level of quality standards which include ISO, Six Sigma or CMM Levels for complete satisfaction of the client matching world class standards in software delivery.

(BTCS 606) SOFTWARE ENGINEERING LAB

COURSE OBJECTIVES

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

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

SYLLABUS

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

After attending Lab sessions, students will

CO1. Able to develop software comprising all phases of SDLC.

CO2. Perform software testing using test cases.

CO3. Implement real world applications using various software paradigms.

HU-251 HUMAN RESOURCE MANAGEMENT

COURSE OBJECTIVES

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

SYLLABUS

Introduction: Introduction to Human Resource Management and its definition, functions of Human Resource Management & its relation to other managerial functions. Nature, Scope and Importance of Human Resource Management in Industry, Role & position of Personnel function in the organization.

Procurement and Placement: Need for Human Resource Planning; Process of Human Resource 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 & Abolition) Act 1970.

Training & Development: Difference between training and Development; Principles of Training; Employee Development; Promotion-Merit v/s seniority Performance Appraisal, Career Development & Planning.

Job analysis & Design: Job Analysis: Job Description & Job Description, Job Specification.

Job Satisfaction: Job satisfaction and its importance; Motivation, Factors affecting motivation, introduction to Motivation Theory; Workers ' Participation, Quality of work life.

The Compensation Function: Basic concepts in wage administration, company's wage policy, Job Evaluation, Issues in wage administration, Bonus & Incentives, Payment of Wages Act-1936, Minimum Wages Act-1961

Integration: Human Relations and Industrial Relations; Difference between Human Relations and Industrial Relations, Factors required for good Human Relation Policy in Industry; Employee Employer relationship Causes and Effects of Industrial disputes; Employees Grievances & their Redressal, Administration of Discipline, Communication in organization, Absenteeism, Labour Turnover, Changing face of the Indian work force and their environment, Importance of collective Bargaining; Role of trade unions in maintaining cordial Industrial Relations.

Maintenance: Fringe & retirement terminal benefits, administration of welfare amenities, Meaning and Importance of Employee Safety, Accidents-Causes & their Prevention, Safety Previsions under the Factories Act 1948; Welfare of Employees and its Importance, Social security, Family Pension Scheme, ESI act 1948, Workmen's Gratuity Act 1972, Future challenges for Human Resource Management.

COURSE OUTCOMES

CO 1: To know and understand the fundamentals of human resource management.

CO 2: To adapt this knowledge to real problems.

CO 3: To develop, implement, and evaluate employee orientation, training, and development programs.

CO 4: To facilitate and support effective employee and labour relations.

BTCS 912 – CLOUD COMPUTING (ELECTIVE-III)

COURSE OBJECTIVES

1. To learn how to use Cloud Services.

2. To implement Virtualization

3. To implement Task Scheduling algorithms.

4. Apply Map-Reduce concept to applications.

SYLLABUS

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

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

CO1. Understanding the key dimensions of the challenge of Cloud Computing
CO2.Compare and Contrast different Cloud Deployment and Service models in order to be able to Assess financial, and technological implications in selecting cloud model for own organization.
CO3. Analyzing the different underlying technologies used inimplementing a Cloud.
CO4. Write comprehensive case studies analysing and contrasting different cloud computing solutions.
CO5. Assessing the need of Security in Cloud Computing.

SEVENTH SEMESTER

BTIT-701 BUILDING ENTERPRISE APPLICATION

COURSE OBJECTIVES

7. To familiarize the students of IT branch in engineering with building an enterprise applications which is required to help them further in IT sector.
8. To discuss the concepts of software processes and software products (deliverables) in IT.
9. To familiarize the students of all branches in engineering with Problem Solving.
10. To familiarize the students with software development skills.
11. To familiarize the students with software testing skills.

SYLLABUS

Unit I: Introduction to enterprise applications and their types, software engineering methodologies, life cycle of raising an enterprise application, introduction to skills required to build an enterprise application, key determinants of successful enterprise applications, and measuring the success of enterprise applications

Unit II: Inception of enterprise applications, enterprise analysis, business modeling, requirements elicitation, use case modeling, prototyping, non functional requirements, requirements validation, planning and estimation

Unit III: Concept of architecture, views and viewpoints, enterprise architecture, logical architecture, technical architecture - design, different technical layers, best practices, data architecture and design – relational, XML, and other structured data representations, Infrastructure architecture and design elements - Networking, Internetworking, and Communication Protocols, IT Hardware and Software, Middleware, Policies for Infrastructure Management, Deployment Strategy, Documentation of application architecture and design

Unit IV: Construction readiness of enterprise applications - defining a construction plan, defining a package structure, setting up a configuration management plan, setting up a development environment, introduction to the concept of Software Construction Maps, construction of technical solutions layers, methodologies of code review, static code analysis, build and testing, dynamic code analysis – code profiling and code coverage

Unit V: Types and methods of testing an enterprise application, testing levels and approaches, testing environments, integration testing, performance testing, penetration testing, usability testing, globalization testing and interface testing, user acceptance testing, rolling out an enterprise application.

COURSE OUTCOMES

CO 5: Students will have sufficient knowledge of Challenges faced during building an enterprise application

CO 6: Students will have sufficient knowledge of Life-cycle of building an enterprise application

CO 7: Students will have sufficient knowledge of Measuring the success of enterprise application

CO 8: Students will have knowledge To define, formulate and analyze a problem

CO 9: Students will have sufficient knowledge of tools used in designing use-case diagrams, DFDs, ER diagrams, as-if and to-be diagrams

CO 10: Students will have sufficient knowledge of how to perform static and dynamic code analysis.

CO 11: Students will have sufficient knowledge t build test-cases for verifying the different functional as well as non-functional requirements.

BTIT-702 SOFTWARE PROJECT MANAGEMENT

COURSE OBJECTIVES:

1. To Understand the fundamental principles of Software Project management & have a good knowledge of responsibilities of project manager
2. To be familiar with various stages of software Project planning.
3. To be familiar with the different methods and techniques used for project management.
4. To have the knowledge about Project Scheduling, tracking, Risk analysis, Quality management and Project Cost estimation using different techniques.

SYLLABUS

Unit-1

Project Management Fundamentals- Basic Definitions, Project Stakeholders and Organizational, Influences on Project Management, Project Management Processes, Project Initiating Processes.

Unit-2

Planning and Resourcing a Project - Identifying Requirements, Creating the Work Breakdown structure, Developing the Project Schedule, Developing a Project Cost Estimate, Planning Quality, Organizing the Project Team, Planning for Potential Risks

Unit-3

Executing and Managing a Project -Project Executing Processes- Acquiring and Developing the Project Team, Managing the Project Team, Managing Stakeholder Expectations, Directing and Managing the Project while assuring Quality.

Unit - 4

Project Monitoring and Controlling Processes - Verifying and Controlling Scope, Managing Schedule and Cost, Controlling Quality, Monitoring and Controlling Risks.

Unit-5

Integrated Change Control, Project Closing Process - Closing a Project

COURSE OUTCOMES

After studying this course:

CO1. The student will be able to understand and practice the process of project management

CO2. The student will be able to develop the scope of work, provide accurate cost estimates and to plan the various activities.

CO3. The student will be able to understand and use risk management analysis techniques that identify the factors that put a project at risk and to quantify the likely effect of risk on project timescales

CO4. The students will be able to identify the resources and people required for a project and to produce a work plan and resource schedule.

(BTCS905) SOFTWARE TESTING AND QUALITY ASSURANCE

COURSE OBJECTIVES

1. Develop methods and procedures for software development that can scale up for large systems and that can be used to consistently produce high-quality software at low cost and with a small cycle time.

2. This course offers a good understanding of the concepts, methods and techniques of software
testing and quality assurance and prepares students to be in a position to develop error free and quality software.

SYLLABUS

Introduction: Overview of Software Engineering, Software Process, Process Models, Overview of Project Management Process and its Phases. (7)

Software Quality Assurance Concepts and Standards: Quality Concepts, Quality Control, Quality Assurance, SQA Activities, Software Reviews, Formal Technical Reviews, Review Guidelines, Software Reliability, Software Safety, Quality Assurance Standards, ISO 9000, ISO 9001:2000, ISO 9126 Quality Factors, CMM, TQM, Six Sigma, SPICE, Software Quality Assurance Metrics. (8)

Risk Management and Change Management: Software Risks, Risk Identification, Risk Projection, Risk Refinement, The RMMM Plan, Software Configuration Management, Baselines, Software Configuration Items, SCM Process: Version Control, Change Control, Configuration Audit, Configuration Management for Web Engineering. (7)

Software Testing: Testing, Verification and Validation, Test Strategies for Conventional and Object Oriented Software, Unit Testing, Integration Testing, Validation Testing, Alpha and Beta Testing, System Testing, Recovery Testing, Security Testing, Stress Testing, Performance Testing, Metrics for Source Code, Metrics for Testing, Debugging Process, Debugging Strategies. (7)

Testing Techniques: Software Testing Fundamentals, Black Box and White Box Testing, Basis Path Testing, Flow Graph Notation, Independent Program Paths, Graph Matrices, Control Structure Testing, Condition Testing, Data Flow Testing, Loop Testing, Graph Based Testing Methods, Equivalence Partitioning,Object Oriented Testing Methods: Applicability of Conventional Test Case Design Methods, Issues in Object Oriented Testing, Fault-Based Testing, Scenario-Based Testing, Random Testing and Partition Testing for Classes, InterClass Test Case Design. (8)

Testing Process and Specialized Systems Testing: Test Plan Development, Requirement Phase, Design Phase and Program Phase Testing, Testing Client/Server Systems, Testing Web based

Systems, Testing Offthe-Shelf Software, Testing in Multiplatform Environment, Testing for Real Time Systems, Testing Security.

COURSE OUTCOMES

CO1. Analyze different approaches to software testing and quality assurance, and select optimal solutions for different situations and project.

CO2. Create test strategies and plans, design test cases, prioritize and execute them

CO3. Manage incidents and risks within a project and generate a risk projection.

CO4. Contribute to efficient delivery of software solutions and implement improvements in the software development processes.

CO5. Compare and Contrast the various activities of Quality Assurance, Quality planning and Quality Control.

BTIT 906 ADVANCED JAVA

OBJECTIVES: This course will equip the students with all the concepts in order to build robust client server applications.

SYLLABUS

Unit I: Introduction to Multithreading and Concurrency in Java

Creating and managing threads in Java, Priority management, Thread synchronization, Inter thread communication, Thread groups and Daemon threads.

Concept of concurrency, task scheduling, Callable and Futures, Synchronizers, Semaphores, Concurrent collections, Atomic variables and Locks.

Unit II: Understanding Input Output Streams

Basic concepts of Stream data, Input Stream hierarchy, Output Stream hierarchy, Understanding of various API’s and methods used or streaming of data. Serialization and security in Serialization.

Unit III: Introduction to Java Data Base Connectivity

Basic concept on Database Connectivity Drivers, Database interaction using Statement Interface, Result set Interface. Prepared Statements and Callable statements. Transaction management.

Unit IV: Java Beans and Generics

Reflection API, Introduction to Java Bean, Java Beans in User Interface, Naming Convention, Importance of Bean Serialization. Introduction to generics, Impolementation of various types of Generics and Concept of Erasure.

Unit V: Annotations

Introduction to Annotations, Build in Annotation Inheritance, Creation of user defined Annotations and Advantages of Annotations

COURSE OUTCOMES

CO1: Basics of Java Programming Language to achieve concurrency and multithreading for high performance.

CO2: Connect to database using Java's JDBC connectivity and performing various operations on database tables, including creating, reading, updating and deleting using the same.

CO3: The concept behind the client/server model with hands-on details in creating a TCP client server application in Java, learning to build client based applications.