PUNJAB TECHNICAL UNIVERSITY

Scheme& Syllabus of B. Tech. Computer Science & Engineering [CSE]

3rd to 5th Semester for affecting Batch 2011

By
Board of Studies Computer Science Engineering/ Information Technology / Computer Applications

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	2
BTCS302	Discrete Structures	3	1	-	40	60	100	2
BTCS303	Digital Circuits & Logic Design	3	1	-	40	60	100	2
BTCS304	Data Structures	3	1	-	40	60	100	2
BTCS305	Object Oriented Programming using C++	3	1	-	40	60	100	2
BTCS306	Data Structures Lab	-	-	4	30	20	50	1
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**	**14**

Fourth Semester **Contact Hours:**

Course Code	Course Name	Load Allocation			Marks Distribution		Total Marks	Credits
		L	T	P	Internal	External		
BTCS401	Operating Systems	3	1	-	40	60	100	2
BTCS402	Mathematics -III	3	1	-	40	60	100	2
BTCS403	Computer Networks-I	3	1	-	40	60	100	2
BTCS404	Microprocessor& Assembly Language Programming	3	1	-	40	60	100	2
BTCS405	System Programming	3	1	-	40	60	100	2
BTCS406	Operating System Lab	-	-	4	30	20	50	1
BTCS407	Computer Networks-I Lab	-	-	2	60	40	100	1
BTCS408	Microprocessor& Assembly Language Programming Lab	-	-	2	30	20	50	1
BTCS409	System Programming Lab	-	-	4	30	20	50	2
	General Fitness	-	-	-	100	-	100	-
	Total	**15**	**5**	**10**	**450**	**400**	**850**	**14**

Punjab Technical University

B.Tech. Computer Science & Engineering (CSE)

Batch 2011

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	RDBMS-I	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-I 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**

Third Semester

BTCS 301 Computer Architecture

PART-A

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]**

PART-B

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]**

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.

BTCS302 Discrete Structures

Objective/s:

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.

PART-A

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. **[7]**

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) **[8]**
3. **Combinatorial Mathematics:** Basic counting principles Permutations and combinations Inclusion and Exclusion Principle Recurrence relations, Generating Function, Application. **[7]**

PART-B

4. **Monoids and Groups:** Groups Semigroups and monoids Cyclic semigraphs and submonoids, Subgroups and Cosets. Congruence relations on semigroups. Morphisms. Normal subgroups. Dihedral groups. **[7]**
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.

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.

BTCS303 Digital Circuits & Logic Design

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

PART-A

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. **[5]**
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. **[5]**

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. **[5]**
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. **[6]**

PART-B

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. **[4]**
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). **[4]**
7. **Signal Conversions:** Analog & Digital signals. A/D and D/A conversion techniques (Weighted type, R-2R Ladder type, Counter Type, Dual Slope type, Successive Approximation type). **[5]**

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.

BTCS 304 Data Structures

PART-A

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]**
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. **[2]**

3. **Arrays:** Linear and multi-dimensional arrays and their representation, operations on arrays, sparse matrices and their storage. **[3]**
4. **Linked List:** Linear linked list, operations on linear linked list, doubly linked list, operations on doubly linked list, application of linked lists. **[4]**
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. **[4]**
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. **[4]**

PART-B

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. **[4]**
8. **Heaps:** Representing a heap in memory, operations on heaps, application of heap in implementing priority queue and heap sort algorithm. **[2]**
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. **[3]**
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. **[3]**
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. **[5]**

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. R. S. Salaria, **Test Your Skills in Data Structures**

BTCS 305 Object Oriented Programming Using C++

PART-A

1. **Object-Oriented Programming Concepts:** Introduction, comparision between procedural programming paradigm and object-oriented proramming 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]**
2. **Standard Input/Output:** Concept of streams, hierarchy of console stream classes, input/output using overloaded operators >> and << and memberv functions of i/o stream classes, formatting output, formatting using ios class functions and flags, formatting using manipulators. **[3]**
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]**
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]**
5. **Constructors and Destructors:** Need for constructors and destructors, copy constructor, dynamic constructors, explicit constructors, destructors, constructors and destructors with static members, initializer lists. **[2]**
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. **[4]**

PART-B

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. **[5]**
8. **Virtual functions & Polymorphism:** Concept of binding - early binding and late binding, virtual functions, pure virtual functions, abstract clasess, virtual destructors. **[3]**
9. **Exception Handling:** Review of traditional error handling, basics of exception handling, exception handling mechanism, throwing mechanism, catching mechanism, rethrowing an exception, specifying exceptions. **[2]**

10. **Templates and Generic Programming:** Template concepts, Function templates, class templates, illustrative examples. **[3]**
11. **Files:** File streams, hierarchy of file stream classes, error handling during file operations, reading/writing of files, accessing records randomly, updating files. **[3]**

Suggested Readings/ Books:

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. R. S. Salaria, **Test Your Skills in Object-Oriented Programming With C++**, Salaria Publishing House.

BTCS306 Data Structures Lab

List of practical exercises, to be implemented using object-oriented approach in C++ Language.

1. Write a menu driven program that implements following operations (using separate functions) on a linear array:

- Insert a new element at end as well as at a given position
- Delete an element from a given 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

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.

BTCS 308 Digital Circuits & Logic Design Lab

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.

BTCS 309 Object Oriented Programming Using C++ Lab

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.

Fourth Semester

BTCS 401 Operating Systems

PART-A

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. **[5]**
2. **Process management:** CPU scheduling, Scheduling Algorithms, PCB, Process synchronization, Deadlocks, Prevention, Detection and Recovery **[5]**
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 **[8]**

PART-B

4. **Device Management:** I/O system and secondary storage structure, Device management policies, Role of I/O traffic controller, scheduler **[5]**
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 **[4]**

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

BTCS402 Mathematics-III

Objective/s and Expected Outcome: 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. **[4]**
2. **Laplace transformations:** Laplace transforms of various standard functions, properties of Laplace transform. **[4]**

3. **Partial Differential Equations:** Formation of Partial Differential Equations, linear Partial Differential Equations, Homogeneous Partial Differential Equations with constant coefficients. **[5]**
4. **Functions of complex variables:** Limits, continuity and derivatives of the function of complex variables, Analytic function, Cauchy- Riemann equations, conjugate functions. **[5]**

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. **[4]**
6. **Differential Equations:** Solutions of Initial values problems using Eulers, modified Eulers method and Runge- kutta (upto fourth order) methods. **[4]**
7. **Probability distribution:** Binomial, Poisson and Normal distribution. **[4]**
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. **[5]**

Suggested Readings/ Books:

1. E. Kreyszig, Advanced Engineering Mathematics, 5th Edition, Wiley Enstern 1985.
2. P. E. Danko, A. G. Popov, T. Y. A. Kaznevnikova, Higher Mathematics in Problems and Exercise, Part 2, Mir Publishers, 1983.
3. Bali, N. P., A Text Book on Engineering Mathematics, Luxmi Pub., New Delhi.

BTCS 403 Computer Networks-I

Objective/s and Expected Outcome: This course provides knowledge about computer network related hardware and software using a layered architecture.

PART-A

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, ISO-OSI reference model, TCP/IP reference model. **[7]**

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. **[6]**

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. **[6]**

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. **[6]**

PART-B

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]**

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. **[3]**

7. **Application Layer**:

World Wide Web (WWW), Domain Name System (DNS), E-mail, File Transfer Protocol (FTP), Introduction to Network security **[2]**

Suggested Readings/ Books:

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.

BTCS404 Microprocessors and Assembly Language Programming

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

PART-A

1. **Introduction:** Introduction to Microprocessors, history, classification, recent microprocessors. **[5]**
2. **Microprocessor Architecture:** 8085 microprocessor Architecture. Bus structure, I/O, Memory &

System buses, concept of address Bus, Data Bus & Control Bus, Synchronous & Asynchronous buses. Instruction execution sequence & Data Flow, Instruction cycle. **[5]**

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. **[6]**

PART-B

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. **[7]**
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. **[8]**
6. **Basic architecture of higher order microprocessors:** Basic introduction to 8086 family, motorola 68000, Pentium processors. **[5]**

Suggested Readings/ Books:

1. **8085 Microprocessor** by Ramesh Gaonkar, 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.

BTCS 405 System Programming

Objective/s and Expected Outcome: This course provides knowledge to design various system programs.

1. **Introduction:** Introduction to system programming and different types of system programs – editors, assemblers, macroprocessors, compilers, linkers, loader, debuggers. **[2]**
2. **Assemblers:** Description of single pass and two pass assemblers, use of data structures like OPTAB and SYMTAB, etc. **[9]**
3. **Macroprocessors:** Description of macros, macro expansion, conditional and recursive macro expansion. **[5]**
4. **Compilers:** Various phases of compiler – lexical, syntax and semantic analysis, intermediate code generation, code optimization techniques, code generation, Case study : LEX and YACC. **[9]**
5. **Linkers and Loaders:** Concept of linking, different linking schemes, concept of loading and various loading schemes. **[5]**
6. **Editors:** Line editor, full screen editor and multi window editor, Case study MS-Word, DOS Editor and vi editor. **[4]**

7. **Debuggers:** Description of various debugging techniques. **[2]**

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.

BTCS 406 Operating System Lab

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.

BTCS 407 Computer Networks-I Lab

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.

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.

References:

Microprocessor by B. Ram, Dhanpat Rai Publications.

BTCS 409 System Programming Lab

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.