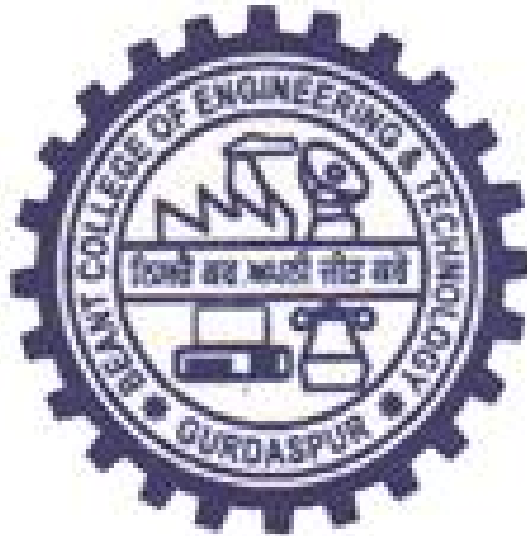


For Batches 2015 & Onwards  
Academic Autonomous Status vide letter No. F22-1/2014 (AC)

## **BEANT COLLEGE OF ENGINEERING & TECHNOLOGY, GURDASPUR**

**Scheme & Syllabus of**  
**B. Tech.**  
**Computer Science Engineering (CSE)**  
**Batch 2015 onwards**



# BEANT

**By**  
**Department of Academics**  
**BEANT COLLEGE OF ENGINEERING & TECHNOLOGY**  
**GURDASPUR**

For Batches 2015 & Onwards  
Academic Autonomous Status vide letter No. F22-1/2014 (AC)  
**Beant College of Engg.& Tech. Gurdaspur**  
B.Tech. Computer Science and Engineering (CSE)

**Third Semester**

**Contact Hours: 26 Hrs.**

Course Code	Course Name	Load Allocation			Marks Distribution		Total Marks	Credits
		L	T	P	Internal	External		
BTCS-301	Computer Architecture	3	1	-	40	60	100	4
BTAM-302	Mathematics-III	3	1	-	40	60	100	4
BTCS-303	Digital Circuits & Logic Design	3	1	-	40	60	100	4
BTCS-304	Data Structures	3	1	-	40	60	100	4
BTCS-305	Object-Oriented Programming using C++	3	1	-	40	60	100	4
BTCS-306	Data Structures Lab	-	-	2	30	20	50	1
BTCS-307	Institutional Practical Training*	-	-	-	60	40	100	2
BTCS-308	Digital Circuits & Logic Design Lab	-	-	2	30	20	50	1
BTCS-309	Object-Oriented Programming using C++ Lab	-	-	2	30	20	50	1
<b>Total:</b>		<b>15</b>	<b>5</b>	<b>6</b>	<b>350</b>	<b>400</b>	<b>750</b>	<b>25</b>

\*The marks will be awarded on the basis of 4-week Institutional Practical training conducted after 2nd semester

**Fourth Semester**

**Contact Hours: 28 Hrs.**

Course Code	Course Name	Load Allocation			Marks Distribution		Total Marks	Credits
		L	T	P	Internal	External		
BTCS-401	Operating systems	3	1	-	40	60	100	4
BTAM-402	Discrete Structures	3	1	-	40	60	100	4
BTCS-403	Computer Networks - I	3	1	-	40	60	100	4
BTCS-404	Microprocessor and Assembly Language Programming	3	1	-	40	60	100	4
BTCS-405	System Programming	3	1	-	40	60	100	4
BTCS-406	Operating System Lab	-	-	2	30	20	50	1
BTCS-407	Computer Networks - I Lab	-	-	2	30	20	50	1
BTCS-408	Microprocessor and Assembly Language Programming Lab	-	-	2	30	20	50	1
BTCS-409	System Programming Lab	-	-	2	30	20	50	1
BTGF-400	General Fitness				100		100	1
<b>Total:</b>		<b>15</b>	<b>5</b>	<b>08</b>	<b>420</b>	<b>380</b>	<b>800</b>	<b>25</b>

**Fifth Semester**

**Contact Hours: 27 Hrs.**

Course Code	Course Name	Load Allocation			Marks Distribution		Total Marks	Credits
		L	T	P	Internal	External		
BTCS-501	Computer Network - II	3	1	-	40	60	100	4
BTCS-502	Relational database Management System	3	1	-	40	60	100	4
BTCS-503	Design & Analysis of Algorithms	3	1	-	40	60	100	4
BTCS-504	Computer Graphics	3	1	-	40	60	100	4
BTXXXXX	Open Elective-I	3	0	-	40	60	100	3
BTCS-506	RDBMS Lab	-	-	2	30	20	50	1
BTCS-507	Computer Networks - II Lab	-	-	2	30	20	50	1
BTCS-508	Design & Analysis of Algorithms Lab	-	-	2	30	20	50	1
BTCS-509	Computer Graphics Lab	-	-	2	30	20	50	1
BTCS-510	Industrial Training*	-	-	-	60	40	100	2
<b>Total:</b>		<b>15</b>	<b>4</b>	<b>08</b>	<b>380</b>	<b>420</b>	<b>800</b>	<b>25</b>

\*The marks will be awarded on the basis of 06 weeks industrial training conducted after 4<sup>th</sup> semester

**Sixth Semester**

**Contact Hours: 29 Hrs.**

Course Code	Course Name	Load Allocation			Marks Distribution		Total Marks	Credits
		L	T	P	Internal	External		
BTCS-601	Programming in Java	3	1	-	40	60	100	4
BTCS-602	Data Mining and Data Warehousing	3	1	-	40	60	100	4
BTCS-603	Software Engineering	3	1	-	40	60	100	4
BTCS-xxx	Elective-I	3	1	-	40	60	100	4
BTXXXXX	Open Elective -II	3	0	-	40	60	100	3
BTCS-604	Data Mining and Data Warehousing Lab	-	-	2	30	20	50	1
BTCS-605	Free/Open Source Software Lab	-	-	2	30	20	50	1
BTCS-606	Software Engineering Lab	-	-	2	30	20	50	1
BTCS-607	Java Programming Lab	-	-	2	30	20	50	1
BTCS-608	Minor Project	-	-	2	30	20	50	1
BTGF-600	General Fitness	-	-	-	100	-	100	1
<b>Total:</b>		<b>15</b>	<b>4</b>	<b>10</b>	<b>450</b>	<b>400</b>	<b>850</b>	<b>25</b>

**Seventh Semester**

**Contact Hours: 27 Hrs.**

Course Code	Course Name	Load Allocation			Marks Distribution		Total Marks	Credits
		L	T	P	Internal	External		
BTCS-701	Artificial Intelligence	3	-	-	40	60	100	3
BTCS-702	Theory of Computation	3	1	-	40	60	100	4
BTCS-703	Computer Peripherals & Interfaces	3	1	-	40	60	100	4
BTCS-704	Major Project	-	-	8	100*	100	200	4
BTCSYYY	Elective-II	3	-	-	40	60	100	3
BTCSZZZ	Elective-III	3	-	-	40	60	100	3
BTCS-705	Artificial Intelligence Lab	-	-	2	30	20	50	1
BTGF-800	General Fitness				100		100	1
<b>Total:</b>		<b>15</b>	<b>2</b>	<b>10</b>	<b>430</b>	<b>420</b>	<b>850</b>	<b>23</b>

**Eighth Semester**

Course Code	Course Name	Marks Distribution		Total Marks	Credits
		Internal	External		
BTCS-801	Industrial Training	450	300	750	24
<b>Total:</b>		<b>450</b>	<b>300</b>	<b>750</b>	<b>24</b>

**Total Credits: 147 (3<sup>rd</sup> to 8<sup>th</sup> Semester) + 50 (1<sup>st</sup> & 2<sup>nd</sup> Semester) = 197**

**Total marks: 4800 (3<sup>rd</sup> to 8<sup>th</sup> Semester) + 1600 (1<sup>st</sup> & 2<sup>nd</sup> Semester) = 6400**

**Elective –I BTXS XXX**

1. BTCS-901 Web Technologies
2. BTCS-902 Mobile Applications Development
3. BTCS-903 Ethical Hacking
4. BTCS-904 Information Security
5. BTCS-905 Simulation and Modeling

**Elective –II BTCS YYY**

1. BTCS-911 Software Testing and Quality Assurance
2. BTCS-912 Object Oriented Analysis and Design
3. BTCS-913 Software Project Management
4. BTCS-914 Business Intelligence
5. BTCS-915 Agile Software Development
6. □□CS-916 Internet Protocols

**Elective -III BTCS ZZZ**

1. BTCS-921 Multimedia and Application
2. BTCS-922 Soft Computing
3. BTCS-923 Cloud Computing
4. BTCS-924 Compiler Design
5. BTCS-925 Big Data
6. BTCS-926 Digital Image Processing
7. BTCS-927 Enterprise Resource Planning
8. BTCS-928 Advanced Operating System

## **List of Open Electives Offered by CSE Department**

Open Elective subject is offered in 5<sup>th</sup> & 6<sup>th</sup> Semesters. The students of any discipline other than CSE may opt any of the following subject as an open elective.

### **Open Electives for 5th Semester:**

1. **BTCS-951** Data Structures
2. **BTCS-952** Web Technologies
3. **BTCS-953** Information Security

### **Open Electives for 6th Semester:**

1. **BTCS-961** Computer Networks-I
2. **BTCS-962** Operating System
3. **BTCS-963** Microprocessor & Assembly Language Programming
4. **BTCS-964** Software Engineering

For Batches 2015 & Onwards  
Academic Autonomous Status vide letter No. F22-1/2014 (AC)

Beant College of Engineering & Technology, Gurdaspur

# *Third Semester*

# BCET

## BTCS-301 Computer Architecture

**L T P**  
**3 1 0**

**Internal Marks: 40**

**External Marks: 60**

**Total Marks: 100**

### **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. Finally the student will be exposed to the recent trends in parallel and distributed computing and multithreaded application.

### **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. (6)

### **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. (5)

### **Design of Control Unit:**

Control memory, design of control unit—microprogrammed, hardwired, and their comparative study. (6)

### **Central Processing Unit:**

General Register Organisation, Stack Organisation, Instruction formats, Addressing Modes, Data transfer and manipulations, Program control, RISC and CISC architecture. (6)

### **Input-Output Organisation:**

Peripheral devices, I/O Interface, asynchronous data transfer, modes of transfer, priority interrupt, DMA, I/O processor, serial communication. (6)

### **Memory Organisation:**

Memory hierarchy, main memory, auxiliary memory, associative memory, cache memory, virtual memory, memory management hardware. (6)

### **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 Stalling, Computer Organization and Architecture, Pearson Education



3. David A Patterson, Computer Architecture, Pearson Education
4. P. Pal Choudhari, Computer Organization and Design, PHI
5. J P Hayes, Computer System Architecture, Pearson Education
6. Kai Hawang, Advanced Computer Architecture, TMH
7. Reiss, Assembly Language and Computer Architecture using C++ and Java, Cengage Learning.

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

1. Understand how computer hardware has evolved to meet the needs of multi processing systems.
2. Understand the major components of a computer like CPU, memory, I/O and storage.
3. Understand the uses for cache memory.
4. Understand a wide variety of memory technologies both internal and external.
5. Understand the role of the operating system in interfacing with the computer hardware.
6. Understand the basic components of the CPU including the ALU and control unit.
7. Have a basic understanding of assembly programming. Students will understand design principles in instruction set design including RISC architectures.
8. Understand parallelism both in terms of a single processor and multiple processors.
9. To have the knowledge of latest topics.

BCET

**BTAM-302 Mathematics-III**

**L T P**  
**3 1 0**

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

**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.

**Fourier series:**

Periodic Functions, Euler's Formula. Even and odd Functions, Half range expansions, Fourier series of different waveforms. (5)

**Laplace transformations:**

Laplace transforms of various standard functions, properties of Laplace transform. (5)

**Partial Differential Equations:**

Formation of Partial Differential Equations, linear Partial Differential Equations, Homogeneous Partial Differential Equations with constant coefficients. (5)

**Functions of complex variables:**

Limits, continuity and derivatives of the function of complex variables, Analytic function, Cauchy-Riemann equations, conjugate functions. (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. (5)

**Differential Equations:**

Solutions of Initial values problems using Eulers, modified Eulers method and Runge- kutta (upto fourth order) methods. (5)

**Probability distribution:**

Binomial, Poisson and Normal distribution. (5)

**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:**

- a. E. Kreyszig, "Advanced Engineering Mathematics", 5<sup>th</sup> Edition, Wiley Enstern 1985.

- b. P. E. Danko, A. G. Popov, T. Y. A. Kaznevnikova, "Higher Mathematics in Problems and Exercise", Part 2, Mir Publishers, 1983.
- c. Bali, N. P., "A Text Book on Engineering Mathematics", Luxmi Pub., New Delhi.
- d. Peter V.O'Neil, "Advanced Engineering Mathematics", Cengage Learning

**COURSE OUTCOMES (CO):** The expected outcomes are:

1. In Mathematics, a transform is usually a technique that converts one type of function into another type presumably easier to solve.
2. It is representation of a function as a series of constants times sine and cosine functions of different frequencies in order to see periodic phenomenon have long fascinating mankind.
3. Mathematical models of physical phenomenon involving more than one independent variable often include partial differential equations. They also arise in such diverse area as epidemiology, traffic flow studies and the analysis of economics.
4. These have not only a rich theory, but the applications are sometimes surprising as the derivative and integral of complex numbers.
5. Primary motivation for studying certain special functions is that they arise in solving ordinary and partial differential equations that model may physical phenomenon. They constitute necessary items in the toolkit of anyone who wishes to understand the work with such models.

BCET

## BTCS-303 Digital Circuits & Logic Design

L T P  
3 1 0

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

### Objective/s and Expected outcome:

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

### Number Systems:

Binary, Octal, Decimal, Hexadecimal. Number base conversions, 1's, 2's, rth's complements, signed Binary numbers. Binary Arithmetic, Binary codes: Weighted BCD, Gray code, Excess 3 code, ASCII – conversion from one code to another. (6)

### 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. (6)

### 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. (6)

### 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)

### 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). (5)

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

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

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

## BTCS-304 Data Structure

**L T P**  
**3 1 0**

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

**Objectives:** The course should provide one with a fairly good concept of the fundamentals of data structures and also of the commonly occurring algorithms. The mathematical model of data is an abstract concept of data such as set, list or graph. To make it useful for problem solving the abstraction is made concrete by going into the data structure of the model- its implementation and associated algorithms. Given a data structure, quite frequently, several alternative algorithms exist for the same operation. Naturally, the question analyzing an algorithm to determine its performance in relation to the other alternatives becomes important. The course should present the general approach towards analyzing and evaluating algorithms and while presenting an algorithm, its analysis should also be included as and when required.

**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. (4)

**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. (3)

**Linked List:**

Linear linked list, operations on linear linked list, doubly linked list, operations on doubly linked list, application of linked lists. (4)

**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)

**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. (3)

**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)

**Heaps:**

Representing a heap in memory, operations on heaps, application of heap in implementing priority queue and heap sort algorithm. (4)

**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. (4)

**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)

**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 algorithm. (4)

**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.
4. Seymour Lipschutz, Data Structures, Schaum's Outline Series, Tata McGraw Hill
5. Kruse, Data Structures & Program Design, Prentice Hall of India.
6. Michael T. Goodrich, Roberto Tamassia, & David Mount, Data Structures and Algorithms in C++ (Wiley India)
7. Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein, Introduction to Algorithms.
8. Ellis Horowitz, Sartaj Sahni, & Dinesh Mehta, Fundamentals of Data Structures in C++.
9. Malik, Data Structures using C++, Cengage Learning.

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

1. Basic ability to analyze algorithms and to determine algorithm correctness and time efficiency class.
2. Master a variety of advanced abstract data type (ADT) and data structures and their implementations.
3. Master different algorithm design techniques (divide and conquer, greedy, etc.)
4. Ability to apply and implement learned algorithm design techniques and data structures to solve problems.

## BTCS-305 Object Oriented Programming Using C++

L T P  
3 1 0

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

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

### **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. (4)

### **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 string class functions and flags, formatting using manipulators. (4)

### **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, initialize lists. (2)

### **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)

### **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)

### **Virtual functions & Polymorphism:**

Concept of binding - early binding and late binding, virtual functions, pure virtual functions, abstract classes, virtual destructors. (4)



**Exception Handling:**

Review of traditional error handling, basics of exception handling, exception handling mechanism, throwing mechanism, catching mechanism, rethrowing an exception, specifying exceptions. (3)

**Templates and Generic Programming:**

Template concepts, Function templates, class templates, illustrative examples. (2)

**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.Farrell- Object Oriented using C++, Cengage Learning.

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

- 1.Understand object-oriented programming features in C++,
- 2.Apply these features to program design and implementation,
- 3.Understand object-oriented concepts and how they are supported by C++,
- 4.Gain some practical experience of C++,
- 5.Understand implementation issues related to object-oriented techniques,
- 6.Understand the role of patterns in object-oriented design

**BTCS-306 Data Structures Lab**

**L T P**  
**0 0 2**

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

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

1. Write a menu driven program that implements following operations (using separate functions) on a linear array
  - Insert a new element at end as well as at a given position
  - Delete an element from a given 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.

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

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

**BTCS-307 Institutional Practical Training**

**L T P**  
**0 0 0**

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

**4-weeks Institutional Practical Training conducted after 2<sup>nd</sup> semester**

**BTCS-308 Digital Circuits & Logic Design Lab**

**L T P**  
**0 0 2**

**Internal Marks: 30**

**External Marks: 20**

**Total Marks: 50**

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

**Implementation all experiments with help of Bread- Board.**

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

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

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

## BTCS-309 Object Oriented Programming Using C++ Lab

L T P  
0 0 2

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

**OBJECTIVES :** Hands on experience of Object Oriented Programming Concepts with C++, Preparedness to study independently any other Object Oriented Programming language and apply to variety of real time problem scenarios

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 (CO):** The student is expected to:

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

For Batches 2015 & Onwards  
Academic Autonomous Status vide letter No. F22-1/2014 (AC)

Beant College of Engineering & Technology, Gurdaspur

# BCET

***Fourth Semester***

## BTCS-401 Operating Systems

L T P  
3 1 0

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

**Objectives:** This course should provide the students with good understanding of Operating System including its architecture and all its components. Good conceptions on all the topics 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.

### **Introduction:**

Operating system, Role of Operating System as resource manager, function of kernel and shell, operating system structures, views of an operating system. (5)

### **Process management:**

CPU scheduling, Scheduling Algorithms, PCB, Process synchronization, Deadlocks, Prevention, Detection and Recovery (6)

### **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 (9)

### **Device Management:**

I/O system and secondary storage structure, Device management policies, Role of I/O traffic controller, scheduler (6)

### **File Management:**

File System Architecture, Layered Architecture, Physical and Logical File Systems, Protection and Security. (6)

Brief study to multiprocessor and distributed operating systems. (5)

### **Case Studies:**

LINUX / UNIX Operating System and Windows based operating systems. Recent trends in operating system. (3)

### **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. 3<sup>rd</sup> Edition
4. Operating System by Madnick Donovan
5. Operating System by Stallings

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

1. Describe, contrast and compare differing structures for operating systems
2. Understand and analyze theory and implementation of: processes, resource control (concurrency etc.), physical and virtual memory, scheduling, I/O and files
3. Study of various operating systems such as Windows, Linux, Unix etc.

BCET

## BTAM-402 Discrete Structures

L T P  
3 1 0

**Internal Marks: 40**

**External Marks: 60**

**Total Marks: 100**

**Objective:** 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.

### **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. (8)

### **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, map) (8)

### **Combinatorial Mathematics:**

Basic counting principles Permutations and combinations. Inclusion and Exclusion Principle Recurrence relations, Generating Function, Application. (8)

### **Monoids and Groups:**

Groups Semigroups and monoids Cyclic semigroups and Subgroups and Cosets. Congruence relations on semigroups. Morphisms. Normal groups. (8)

### **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. (8)

### **Suggested Readings/ Books:**

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

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

1. Be familiar with constructing proofs.
2. Be familiar with elementary formal logic.
3. Be familiar with set algebra.
4. Be familiar with combinatorial analysis.
5. Be familiar with recurrence relations.
6. Be familiar with graphs and trees, relations and functions, and finite automata.
7. Be exposed to the strategies for compare relative efficiency of algorithms.

## BTCS-403 Computer Networks–I

L T P  
3 1 0

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

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

### **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. (6)

### **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)

### **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)

### **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)

### **Network Layer:**

Design issues, IPv4 classful and classless addressing, subnetting, Routing algorithms: distance vector and link state routing, Congestion control: Principles of Congestion Control, Congestion prevention policies, Leaky bucket and token bucket algorithms. (6)

### **Transport Layer:**

Elements of transport protocols: addressing, connection establishment and release, flow control and buffering, multiplexing and de-multiplexing, crash recovery, introduction to TCP/UDP protocols and their comparison. (5)

### **Application Layer:**

World Wide Web (WWW), Domain Name System (DNS), E-mail, File Transfer Protocol (FTP), Introduction to Network security (5)

**Suggested Readings/ Books:**

- 1.Computer Networks, 4<sup>th</sup> Edition, Pearson Education by Andrew S. Tanenbaum
- 2.Data Communication & Networking, 4th Edition, Tata McGraw Hill. By Behrouz A. Forouzan.
- 3.Computer Networking, 3<sup>rd</sup> 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.
- 5.Guide to Networking Essentials, 5<sup>th</sup> Edition, Cengage Learning by Greg Tomsho,
- 6.Handbook of Networking, Cengage Learning by Michael W. Graves.

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

1. Know the various Computer Networks
2. Understand the various transmission medias for communication;
3. Understand the encoding & decoding techniques;
4. Understand modulation and Demodulation;
5. Know multiplexing and demultiplexing techniques;

BCET

## BTCS-404 Microprocessors and Assembly Language Programming

L T P  
3 1 0

**Internal Marks: 40**

**External Marks: 60**

**Total Marks: 100**

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

**Introduction:** Introduction to Microprocessors, history, classification, recent microprocessors. (6)

### **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. (6)

### **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. (7)

### **Instruction set & Assembly Languages Programming:**

Introduction, instruction & dataformats, addressing modes, status flags, 8085 instructions, Data transfer operations, Arithmetic operations, Logical operations, Branch operations. (7)

### **Case structure & Microprocessor application:**

Interfacing of keyboards and seven segment LEDdisplay, Microprocessor controlled temperature system (MCTS), Study of traffic light system, stepper motor controller, Microprocessor based micro computers. (7)

**Basic architecture of higher order microprocessors:** Basic introduction to 8086 family, Motorola 68000, Pentium processors. (7)

### **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 (CO):** The expected outcomes are:

1. Students should be able to solve basic binary math operations using the microprocessor.
2. Students should be able to demonstrate programming proficiency using the various addressing modes and data transfer instructions of the target microprocessor.
3. Students should be able to program using the capabilities of the stack, the program counter, and the status register and show how these are used to execute a machine code program.
4. Students should be able to apply knowledge of the microprocessor's internal registers and operations by use of a PC based microprocessor simulator.
5. Students should be able to write assemble assembly language programs, assemble into machine a cross assembler utility and download and run their program on the training boards.

## BTCS-405 System Programming

L T P  
3 1 0

**Internal Marks: 40**

**External Marks: 60**

**Total Marks: 100**

**Objective:** This course provides knowledge to design various system programs.

### **Introduction:**

Introduction to system programming and different types of system programs—editors, assemblers, macro-processors, compilers, linkers, loader, debuggers. (5)

### **Assemblers:**

Description of single pass and two pass assemblers, use of data structures like OPTAB and SYMTAB, etc. (8)

### **Macroprocessors:**

Description of macros, macro expansion, conditional and recursive macro expansion. (6)

### **Compilers:**

Various phases of compiler—lexical, syntax and semantic analysis, intermediate code generation, code optimization techniques, code generation, Case study : LEX and YACC. (8)

### **Linkers and Loaders:**

Concept of linking, different linking schemes, concept of loading and various loading schemes. (6)

### **Editors:**

Line editor, full screen editor and multi window editor, Case study MS-Word, DOSEditor and vi editor. (4)

### **Debuggers:**

Description of various debugging techniques. (3)

### **Suggested Readings/ Books:**

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

### **COURSE OUTCOMES (CO):** The expected outcomes are:

1. Master in using the C/C++ programming language, its constructs and grammar, to create system software.
2. Master in the usage of makefiles, linking, object files, loading, symbol resolution, shared and static libraries, debugging, and execution of system programs.
3. Be familiar with basic UNIX OS concepts such as: process, program, process groups, signals, running programs, process control, address space, user and kernel modes, system calls, and context switching.
4. Be familiar with using thread execution models (e.g. Posix threads).
5. Be familiar with different batch processing systems.

## BTCS-406 Operating System Lab

L T P  
0 0 2

**Internal Marks: 30**

**External Marks: 20**

**Total Marks: 50**

**Objective:** This course provides knowledge of different operating systems.

1. Installation Process of various operating systems;

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

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

4. Shell Programming: Basic of shell programming, various types of shell, Shell Programming in bash, conditional & looping statement, case statements, parameter passing and arguments, shell variables, shell keywords, creating shell programs for automate system tasks, report printing.

**COURSE OUTCOMES (CO):** The expected outcomes are:

1. Installation of Operating Systems.
2. Learning of Shell Programming
3. To understand the high-level structure of the Linux kernel both in concept and source code
4. To acquire a detailed understanding of one aspect (the scheduler) of the Linux kernel

**BTCS-407 Computer Networks-I Lab**

**L T P**  
**0 0 2**

**Internal Marks: 30**

**External Marks: 20**

**Total Marks: 50**

**Objective:** To understand the basic concepts of networking.

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

**Course Outcome (CO):** Students are expected to:

1. Know the various components and devices of networking;
2. Know about communication medias;
3. Understand networking topologies;
4. Know networking protocols;
5. Know network classes;



**BTCS-408 Microprocessor and Assembly Language Programming Lab**

**L T P**  
**0 0 2**

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

**Objective:** To understand the basic concepts of microprocessors and assembly language programming.

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 Outcome (CO):** Students are expected to:

1. Know 8085 Microprocessor in detail like its PIN structure;
2. Know various number systems;
3. Arithmetic operations in various numbers systems
4. To Understand 8086 Microprocessor in detail.

**BTCS-409 System Programming Lab**

**L T P**  
**0 0 2**

**Internal Marks: 30**

**External Marks: 20**

**Total Marks: 50**

**Objective:** To understand the basic concepts of system programming language.

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 Outcome (CO):** Students are expected to:

1. Create various kinds of interfaces like pop-up menu, drop-down menu, form-based;
2. Practice of writing assembly language program to implement various data structures like symbol table, literal table etc;
3. Know the use of debug command;
4. Know use of lax and yacc tool

---

**BTGF-400 General Fitness**

**Internal Marks: 100**

**External Marks: 0**

**Total Marks: 100**

**L T P**

**0 0 0**

Beant College of Engineering & Technology, Gurdaspur

# BCET

***Fifth Semester***

## BTCS-501 Computer Networks –II

<b>Internal Marks: 40</b>	<b>L</b>	<b>T</b>	<b>P</b>
<b>External Marks: 60</b>	<b>3</b>	<b>1</b>	<b>0</b>
<b>Total Marks: 100</b>			

**Objectives:** The objective of the course is to offer good understanding of the concepts of network security, wireless, Adhoc and various emerging network technologies.

**Network Security:** Fundamentals of network security, Basics of IPv6, IPsec: overview of IPsec, IP and Ipv6, Authentication header (AH), Encapsulating Security Payload (ESP). [7]

**Encryption/Decryption techniques:** AES, DES Algorithms, Principles of Public Key Cryptosystems, The RSA Algorithms, Key Management, Diffie Hellman Key Exchange. [7]

**Adhoc and Wireless networks:** Features, advantages and applications of adhoc networks, Adhoc versus Cellular networks, Network architecture, Protocols: MAC protocols, Routing protocols, Technologies., Wireless Communication Systems, Examples of wireless communication systems, 2G Cellular networks. [8]

**3G wireless networks:** wireless local loop (WLL), Local Multipoint Distribution System (LMDS), Wireless local Area Networks (WLANs), Bluetooth and Personal Area Networks. [6]

**Wireless System Design:** Introduction, Frequency reuse, channel assignment strategies, handoff strategies, interference and system capacity, improving coverage and capacity in cellular systems. [6]

**Servers:** DHCP, DNS, FTP, SMTP, Mail Server, Web Server [6]

### Suggested Books:

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

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

1. Know the fundamentals of security of computer networks
2. Understand the key management;
3. Understand the basic features of wireless ad hoc networks;
4. Understand FDMA, TDMA and CDMA technologies;
5. Know wireless LAN, WLAN, PAN and Bluetooth networks;
6. Understand wireless system design.

BCET

### **BTCS-502 Relational Database Management System**

<b>Internal Marks: 40</b>	<b>L</b>	<b>T</b>	<b>P</b>
<b>External Marks: 60</b>	<b>3</b>	<b>1</b>	<b>0</b>
<b>Total Marks: 100</b>			

**Objectives:** This course offers a good understanding of database systems concepts and prepares the student to be in a position to use and design databases for different applications.

**Introduction to Database Systems:** File Systems Versus a DBMS, Advantages of a DBMS, Describing and Storing Data in a DBMS, Database System Architecture, DBMS Layers, Data independence. [4]

**Physical Data Organization:**File Organization and Indexing, Index Data Structures, Hashing, B-trees, Clustered Index, Sparse Index, Dense Index, Fixed length and Variable Length Records. [6]

**Data Models:**Relational Model, Network Model, Hierarchical Model, ER Model: Entities, Attributes and Entity Sets, Relationships and Relationship Sets, Constraints, Weak Entities, Class Hierarchies, Aggregation, Conceptual Database Design with the ER Model, Comparison of Models. [5]

**The Relational Model:**Introduction to the Relational Model, ER to Relational Model Conversion, Integrity Constraints over Relations, Enforcing Integrity Constraints, Relational Algebra, Relational Calculus, Querying Relational Data. [5]

**Relational Query Languages:**SQL: Basic SQL Query, Creating Table and Views, SQL as DML, DDL and DCL, SQL Algebraic Operations, Nested Queries, Aggregate Operations, Cursors, Dynamic SQL, Integrity Constraints in SQL, Triggers and Active Database, Relational Completeness, Basic Query Optimization Strategies, Algebraic Manipulation and Equivalences. [5]

**Database Design:**Functional Dependencies, Reasoning about Functional Dependencies, Normal Forms, Schema Refinement, First, Second and Third Normal Forms, BCNF, Multi-valued Dependency, Join Dependency, Fourth and Fifth Normal Forms, Domain Key Normal Forms, Decompositions. [5]

**Transaction Management:**ACID Properties, Serializability, Two-phase Commit Protocol, Concurrency Control, Lock Management, Lost Update Problem, Inconsistent Read Problem , Read-Write Locks, Deadlocks Handling, 2PL protocol. [5]

**Database Protection:**Threats, Access Control Mechanisms, Discretionary Access Control, Grant and Revoke, Mandatory Access Control, Bell LaPadula Model, Role Based Security, Firewalls, Encryption and Digital Signatures. [5]

**Suggested Books:**

1. Ramez Elmasri, Shamkant Navathe ,Fundamentals of Database Systems, Fifth Edition, Pearson Education, 2007.
2. C.J. Date , An Introduction to Database Systems, Eighth Edition, Pearson Education
3. Alexis Leon, Mathews Leon , Database Management Systems, Leon Press.
4. S. K. Singh, Database Systems Concepts, Design and Applications, Pearson Education.
5. Raghu Ramakrishnan, Johannes Gehrke, Database Management Systems, Tata McGraw
6. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, Database System Concepts, Tata McGraw-Hill.

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

1. Understand the various users of database systems;
2. Understand the architecture of database systems;
3. Understand the various data models;
4. know functional dependencies and various normal forms;
5. Understand transaction Processing Concepts,

### BTCS-503 Design & Analysis of Algorithms

<b>Internal Marks: 40</b>	<b>L</b>	<b>T</b>	<b>P</b>
<b>External Marks: 60</b>	<b>3</b>	<b>1</b>	<b>0</b>
<b>Total Marks: 100</b>			

**Objective:** To learn the ability to distinguish between the tractability and intractability of a given computational problem. To be able to devise fast and practical algorithms for real-life problems using the algorithm design techniques and principles learned in this course.

**Introduction:** What is an algorithm ? Time and space complexity of an algorithm. Comparing the performance of different algorithms for the same problem. Different orders of growth. Asymptotic notation. Polynomial vs. Exponential running time. [6]

**Basic Algorithm Design Techniques.** Divide-and-conquer, greedy, randomization, and dynamic programming. Example problems and algorithms illustrating the use of these techniques. [5]

**Graph Algorithms.** Graph traversal: breadth-first search (BFS) and depth-first search (DFS). Applications of BFS and DFS. Topological sort. Shortest paths in graphs: Dijkstra and Bellman-Ford. Minimum spanning trees. [5]

**Sorting and searching.** Binary search in an ordered array. Sorting algorithms such as Merge sort, Quick sort, Heap sort, Radix Sort, and Bubble sort with analysis of their running times. Lower bound on sorting. Median and order statistics. [6]

**NP-completeness.** Definition of class NP. NP-hard and NP-complete problems. 3SAT is NP-complete. Proving a problem to be NP-complete using polynomial-time reductions. Examples of NP-complete problems. [6]

**Coping with NP-completeness.** Approximation algorithms for various NP-complete problems. [6]

**Advanced topics.** Pattern matching algorithms : Knuth-Morris-Pratt algorithm. Algorithms in Computational Geometry : Convex hulls. Fast Fourier Transform (FFT) and its applications. Integer and polynomial arithmetic. Matrix multiplication : Strassen's algorithm. [6]

#### Suggested Books:

1. Algorithm Design by J. Kleinberg and E. Tardos.
2. Introduction to Algorithms by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein.



3. Algorithms by S. Dasgupta, C.H. Papadimitriou, and U.V. Vazirani.
4. Algorithm Design: Foundations, Analysis, and Internet Examples by Michael T. Goodrich and Roberto Tamassia.
5. The Design and Analysis of Computer Algorithms by A. V. Aho, J. E. Hopcroft, and J. D. Ullman.
6. The Art of Computer Programming, Volumes 1, 2, and 3, by Donald Knuth.

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

1. Understand the various models of computation;
2. Do the algorithm analysis and understand the various algorithms design techniques;
3. Know the various sorting and searching techniques; and
4. Know various problem classes.

BCET

### BTCS-504 Computer Graphics

<b>Internal Marks:</b>	<b>40</b>	<b>L</b>	<b>T</b>	<b>P</b>
<b>External Marks:</b>	<b>60</b>	<b>3</b>	<b>1</b>	<b>0</b>
<b>Total Marks:</b>	<b>100</b>			

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

**Introduction:** Computer Graphics and its applications, Elements of a Graphics Systems: Video Display Devices, Raster Scan Systems, Random Scan Systems, Input devices. [5]

**Basic Raster Graphics:** Scan conversion- Point plot technique, Line drawing, Circle generating and Ellipse generating algorithms. [5]

**Two-dimensional Geometric Transformations :** Basic Transformations- Translation, Rotation and Scalling, Matrix Representation and Homogeneous Coordinates, Composite Transformations, Reflection and Shearing transformations. [5]

**Clipping:** Window to viewport transformation, Clipping Operations- Point Clipping, Line Clipping, Polygon Clipping and Text Clipping. [5]

**Filling Techniques:** Scan line algorithms, Boundary-fill algorithm, Flood-fill algorithm, Edge fill and fence fill algorithms, [5]

**Elementary 3D Graphics:** Plane projections and its types, Vanishing points, Specification of a 3D view. [5]

**Visibility:** Image and object precision, Hidden edge/surface removal or visible edge/surface determination techniques; z buffer algorithms, Depth sort algorithm, Scan line algorithm and Floating horizon technique. [5]

**Advance Topics:** Introduction of Rendering, Raytracing, Antialiasing, Fractals, Gourard and Phong shading. [5]

**Suggested Books:**

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

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

1. Understand the various elements of Graphics;
2. Raster graphics concepts;
3. Understand the filling, clipping, geometric manipulation techniques;
4. Know the 3D graphics; and
5. Know various advanced issues like fractal and ray tracing etc.

BCET

**BTXXXXX Open Elective-I**

<b>Internal Marks:</b>	<b>40</b>	<b>L</b>	<b>T</b>	<b>P</b>
<b>External Marks:</b>	<b>60</b>	<b>3</b>	<b>1</b>	<b>0</b>
<b>Total Marks:</b>	<b>100</b>			

**BTCS-506 RDBMS LAB**

<b>Internal Marks:</b>	<b>30</b>	<b>L</b>	<b>T</b>	<b>P</b>
<b>External Marks:</b>	<b>20</b>	<b>0</b>	<b>0</b>	<b>2</b>
<b>Total Marks:</b>	<b>50</b>			

**Objective:** This practical will enable students to retrieve data from relational databases using SQL. Students will also learn about triggers, cursors, stored procedures etc.

1. Introduction to SQL and installation of SQL Server / Oracle.
2. Data Types, Creating Tables, Retrieval of Rows using Select Statement, Conditional Retrieval of Rows, Alter and Drop Statements.
3. Working with Null Values, Matching a Pattern from a Table, Ordering the Result of a Query, Aggregate Functions, Grouping the Result of a Query, Update and Delete Statements.
4. Set Operators, Nested Queries, Joins, 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.

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

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

1. understand the DDL, DML and DCL statements;
2. know the installation of SQL Server/Oracle RDBMSs;
3. use of queries;
4. use of Grant and Revoke commands;
5. use in-built functions, stored procedures, exception handling, triggers and cursors
6. know the nested queries.

### **BTCS-507 Computer Networks – II LAB**

<b>Internal Marks:</b>	<b>30</b>	<b>L</b>	<b>T</b>	<b>P</b>
<b>External Marks:</b>	<b>20</b>	<b>0</b>	<b>0</b>	<b>2</b>
<b>Total Marks:</b>	<b>50</b>			

**Objectives:** The objective of this practical is to offer good understanding of the concepts of wired networks and wireless networks.

1. To configure the IP address for a computer connected to LAN and to configure network parameters of a web browser for the same computer.
2. To plan IPv6 address scheme for a local area network comprising of 'n' terminals.
3. To develop programs for implementing / simulating Encryption Algorithms.
4. To implement the Mail Server in LINUX.
5. To implement the Web Server in LINUX.
6. To implement the FTP Server.
7. To set up the boot server : DHCP and NIS

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

1. Know the fundamentals of computer networks
2. Understand the IPv6 address scheme;
3. Understand the basic features of wireless ad hoc networks;
4. Understand open source packet capture software;
5. Know wireless LAN, WLAN, PAN and Bluetooth networks;
6. Understand wireless system design.

### BTCS-508 Design & Analysis of Algorithms Lab

<b>Internal Marks:</b>	<b>30</b>	<b>L</b>	<b>T</b>	<b>P</b>
<b>External Marks:</b>	<b>20</b>	<b>0</b>	<b>0</b>	<b>2</b>
<b>Total Marks:</b>	<b>50</b>			

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

1. Code and analyze to compute the greatest common divisor (GCD) of two numbers.
2. Code and analyze to find the median element in an array of integers.
3. Code and analyze to find the majority element in an array of integers.
4. Code and analyze to sort an array of integers using Heap sort.
5. Code and analyze to sort an array of integers using Merge sort.
6. Code and analyze to sort an array of integers using Quick sort.
7. Code and analyze to find the edit distance between two character strings using dynamic programming.
8. Code and analyze to find an optimal solution to weighted interval scheduling using dynamic programming.
9. Code and analyze to find an optimal solution to matrix chain multiplication using dynamic programming.
10. Code and analyze to do a depth-first search (DFS) on an undirected graph. Implementing an application of DFS such as (i) to find the topological sort of a directed acyclic graph, OR (ii) to find a path from source to goal in a maze.
11. Code and analyze to do a breadth-first search (BFS) on an undirected graph. Implementing an application of BFS such as (i) to find connected components of an undirected graph, OR (ii) to check whether a given graph is bipartite.
12. Code and analyze to find shortest paths in a graph with positive edge weights using Dijkstra's algorithm.
13. Code and analyze to find shortest paths in a graph with arbitrary edge weights using Bellman-Ford algorithm.
14. Code and analyze to find the minimum spanning tree in a weighted, undirected graph.
15. Code and analyze to find all occurrences of a pattern P in a given string S.

16. Code and analyze to multiply two large integers using Karatsuba algorithm.
17. Code and analyze to compute the convex hull of a set of points in the plane.
18. (Mini-project Topic) Program to multiply two polynomials using Fast Fourier Transform.

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

1. Write code for different methods of computation;
2. Practice of writing program different searching and sorting algorithms and do analysis of the same; and
3. Coding in dynamic programming.

BCET

### **BTCS-509 Computer Graphics Lab**

<b>Internal Marks:</b>	<b>30</b>	<b>L</b>	<b>T</b>	<b>P</b>
<b>External Marks:</b>	<b>20</b>	<b>0</b>	<b>0</b>	<b>2</b>
<b>Total Marks:</b>	<b>50</b>			

**Objective:** To implement various algorithms for different drawing objects.

1. To plot a point (pixel) on the screen.
2. To draw a straight line using DDA Algorithm.
3. To draw a straight line using Bresenham's Algorithm.
4. Implementation of mid-point circle generating Algorithm.
5. Implementation of ellipse generating Algorithm.
6. To translate an object with translation parameters in X and Y directions.
7. To scale an object with scaling factors along X and Y directions.
8. To rotate an object with a certain angle about origin.
9. Perform the rotation of an object with certain angle about an arbitrary point.
10. To perform composite transformations of an object.
11. To perform the reflection of an object about major axis.
12. To clip line segments against windows using Cohen Sutherland Algorithm.
13. Perform the polygon clipping against windows using Sutherland Hodgeman technique.
14. Fill a rectangle with a specified color using scan line algorithm.
15. Implementation of flood-fill and boundary-fill algorithms.

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

1. Do operation on line segments;
2. Do clipping;
3. Implement fill polygon algorithms;
4. Demonstrate the properties of the Bezier curve.



**BTCS-510 Industrial Training**

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

**6-weeks Industrial Training conducted after 4th semester**

---

BCET

Beant College of Engineering & Technology, Gurdaspur

# BCET

*Sixth Semester*

## BTCS-601 Programming in Java

<b>Internal Marks:</b>	<b>40</b>	<b>L</b>	<b>T</b>	<b>P</b>
<b>External Marks:</b>	<b>60</b>	<b>3</b>	<b>1</b>	<b>0</b>
<b>Total Marks:</b>	<b>100</b>			

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

**Overview of Java:** Object oriented programming, paradigms, abstraction, OOP principles, Java class libraries [5]

**Date types, Variables and Arrays:** Integers, floating-point types, characters, Boolean, Iterates, Variable, Data types and casting, automatic type promotion in expressions, arrays. [4]

**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. [3]

**Introduction to Classes:** Class fundamentals, declaring object reference variable, Introducing methods, constructors, this keyword, garbage collection, the finalize () method. [2]

**Methods and Classes:** Overloading methods, using objects as parameters, recursion. [3]

**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. [4]

**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. [4]

**Multithreaded Programming:** The Java thread model, the main thread, creating thread, creating multiple threads, using is alive () and join (), Thread priorities, synchronization, Inter thread communications, suspending resuming and stopping threads. [5]

**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. [4]

**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. [3]

**Networking:** Networking basics, Java and the Net, TCP/IP Client Sockets URL, URL Connection, TCP/IP Server Sockets, Database connectivity. [3]

**Suggested Books**

1. Herbert Schildt, The Complete Reference Java2, McGraw-Hill.
2. Joyce Farrell, Java for Beginners, Cengage Learning.
3. Deitel and Deitel, Java: How to Program, 6th Edition, Pearson Education.
4. James Edward Keogh, Jim Keogh, J2EE: The complete Reference, McGraw-Hill
5. Khalid A. Mughal, Torill Hamre, Rolf W. Rasmussen, Java Actually, Cengage Learning.
6. Shirish Chavan, Java for Beginners, 2<sup>nd</sup> Edition, Shroff Publishers.

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

1. Understand the concepts of OOP & Java language;
2. Know the various techniques such as abstraction, inheritance, and polymorphism;
3. Know the advanced features of Java language; and
4. Understand the Applets & Networking concepts.

BCET

## **BTCS-602 Data Mining and Data Warehousing**

<b>Internal Marks:</b>	<b>40</b>	<b>L</b>	<b>T</b>	<b>P</b>
<b>External Marks:</b>	<b>60</b>	<b>3</b>	<b>1</b>	<b>0</b>
<b>Total Marks:</b>	<b>100</b>			

**Objectives:** This course offers a good understanding of Data warehousing and Data Mining concepts and technologies. It prepares the student to be in a position to use and design data warehouse and application of data mining techniques.

**DATA WAREHOUSING:** Data warehousing Components, Building a Data warehouse, Mapping the Data Warehouse to a Multiprocessor Architecture, DBMS Schemas for Decision Support, Data Extraction, Cleanup, and Transformation Tools, Metadata. [6]

**BUSINESS ANALYSIS:** Reporting and Query tools and Applications, Tool Categories, The Need for Applications, Online Analytical Processing (OLAP), Multidimensional Data Model, OLAP Guidelines, Multidimensional versus Multirelational OLAP, Categories of Tools, OLAP Tools and the Internet. [7]

**DATA MINING:** Introduction – Data – Types of Data – Data Mining Functionalities – Interestingness of Patterns – Classification of Data Mining Systems – Data Mining Task Primitives – Integration of a Data Mining System with a Data Warehouse – Issues –Data Preprocessing. [7]

**ASSOCIATION RULE MINING AND CLASSIFICATION:** Mining Frequent Patterns, Associations and Correlations, Mining Methods, Mining Various Kinds of Association Rules, Correlation Analysis, Constraint Based Association Mining, Classification and Prediction, Basic Concepts, Decision Tree Induction, Bayesian Classification, Rule Based Classification, Classification by Back propagation, Support Vector Machines, Associative Classification, Lazy Learners, Other Classification Methods, Prediction [10]

**CLUSTERING AND APPLICATIONS AND TRENDS IN DATA MINING:** Cluster Analysis, Types of Data, Categorization of Major Clustering Methods, K-means, Partitioning Methods, Hierarchical Methods, Density-Based Methods, Grid Based Methods, Model-Based Clustering Methods, Clustering High Dimensional Data, Constraint-Based Cluster Analysis, Outlier Analysis, Data Mining Applications. [10]

### **Suggested Books:**

1. Alex Berson and Stephen J. Smith, “Data Warehousing, Data Mining & OLAP”, Tata McGraw – Hill Edition, Tenth Reprint 2007.
2. Jiawei Han and Micheline Kamber, “Data Mining Concepts and Techniques”, Second Edition, Elsevier, 2007.
3. Introduction to Data Mining by A. Dunham

4. Pang-Ning Tan, Michael Steinbach and Vipin Kumar, “ Introduction To Data Mining”, Person Education, 2007.
5. K.P. Soman, Shyam Diwakar and V. Ajay “, Insight into Data mining Theory and Practice”, Easter Economy Edition, Prentice Hall of India, 2006.
6. G. K. Gupta, “ Introduction to Data Mining with Case Studies”, Easter Economy Edition, Prentice Hall of India, 2006.
7. Daniel T.Larose, “Data Mining Methods and Models”, Wile-Interscience, 2006.

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

1. understand basic and advanced concepts of data warehousing and data mining;
2. understand the various applications of Data Mining.

BCET

### BTCS-603 Software Engineering

<b>Internal Marks:</b>	<b>40</b>	<b>L</b>	<b>T</b>	<b>P</b>
<b>External Marks:</b>	<b>60</b>	<b>3</b>	<b>1</b>	<b>0</b>
<b>Total Marks:</b>	<b>100</b>			

**Objective:** To understand the basic and advanced concepts of software engineering.

**Evolution and impact of Software engineering**, software life cycle models: Waterfall, prototyping, Evolutionary, and Spiral models. Feasibility study, Functional and Non-functional requirements, Requirements gathering, Requirements analysis and specification [10]

**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. [9]

**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. [9]

**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. [12]

#### Suggested Books:

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

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

1. understand the software principles
2. understand the various phases of SDLC life cycle;
3. know the software project management;
4. understand the quality management like CMMI and Six Sigma;
5. Understand the implementation of software in 3GL and \$GL environment.

# *Elective – I*

# BCET



## BTCS-901 Web Technologies (Elective-I)

<b>Internal Marks: 40</b>	<b>L</b>	<b>T</b>	<b>P</b>
<b>External Marks: 60</b>	<b>3</b>	<b>1</b>	<b>0</b>
<b>Total Marks: 100</b>			

**Objective :** To understand the concept of web programming

**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. (2)

**HTML-5 and CSS:** Introduction to HTML, Lists, adding graphics to HTML page, creating tables, linking documents, forms, DHTML and Cascading Style sheets, Responsive Web Design. (7)

**Java Script:** Introduction, programming constructs: variables, operators and expressions, conditional checking, functions and dialog boxes, JavaScript DOM, creating forms, introduction to Cookies, Lightbox. (5)

**Introduction** to JQuery, JSON, Angular JS, Node.JS, Wordpress, DSN, API's. (8)

**BOOTSTRAP:** Introduction to Grid System, Topography, Tables, Images, Themes, Forms, Inputs. (4)

**XML:** Why XML, XML syntax rules, XML elements, XML attributes, XML DTD displaying XML with CSS (2)

**AJAX:** Introduction, HTTP request, XMLHttpRequest, AJAX Server Script. (2)

**PHP:** Introduction, syntax, statements, operators, sessions, E-mail, PHP and MySQL. (10)

### Suggested Books:

1. HTML,CSS, JavaScript and PHP, Wiley India Textbooks.
2. Ivan Bayross: Web Enabled Commercial Application
3. Deitel,Deitel, Nieto, and Sandhu: XML How to Program, Pearson Education.
4. Online Free Resources from the Internet

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

1. Understand the basics of Internet.
2. Know the tags of HTML, Javascript etc.
3. Understand the Bootstrap, XML, AJAX and PHP Programming.

**BTCS-902 Mobile Applications Development (Elective-I)**

<b>Internal Marks:</b>	<b>40</b>	<b>L</b>	<b>T</b>	<b>P</b>
<b>External Marks:</b>	<b>60</b>	<b>3</b>	<b>1</b>	<b>0</b>
<b>Total Marks:</b>	<b>100</b>			

**Objective:** To understand the development environment for mobile applications.

**Introduction:** Mobile operating system, Operating system structure, Constraints and Restrictions, Hardware configuration with mobile operating system, Features: Multitasking Scheduling, Memory Allocation, File System Interface, Keypad Interface, I/O Interface, Protection and Security, Multimedia features. [6]

**Introduction to Mobile development IDE's,** Introduction to Worklight basics, Optimization, pages and fragments , Writing a basic program- in Worklight Studio, Client technologies, Client side debugging, Creating adapters, Invoking adapters from Worklight Client application, Common Controls, Using Java in adapters, Programming exercise with Skins, Understanding Apache Cordova, Offline access, Encrypted cache deprecated, Using JSONStore [5]

**Understanding Apple iOS development:** Android development, Shell Development, Creating Java ME application, Exploring the Worklight Server, Working with UI frameworks, Authentication, Push notification, SMS Notifications, Globalization, WebView overlay , Creating Authentication application: development for Apple iOS by using a login module, Device Analytics, Worklight Server Administration [5]

**Windows Phone:** Introduction to Windows Phone, Architecture, memory management, communication protocols, application development methods, deployment. [5]

**Case Study:** Design and development of Application using mobile application development platforms e.g. WorkLight, Kendo, Appcon, Xcode, Xpages [5]

**Android:** Introduction to Android, Architecture, memory management, communication protocols, application development methods, deployment [5].

**Case Study:** Design and development of Application using mobile application development platforms e.g. WorkLight, Kendo, Appcon, Xcode, Xpages [5]

**iOS:** Introduction to iOS, Architecture, memory management, communication protocols, application development methods, deployment. [5]

**Case Study:** Design and development of Application using mobile application development platforms e.g. WorkLight, Kendo, Appcon, Xcode, Xpages [4]

**Suggested Books:**

1. Anubhav Pradhan, Anil V Deshpande, “ Mobile Apps Development” Edition: I
2. Jeff McWherter, Scott Gowell “Professional Mobile Application Development”, John Wiley & Sons, 2012.
3. Barry Burd, “Android Application Development All in one for Dummies”, Edition: I
4. Teach Yourself Android Application Development In 24 Hours, Edition: I, Publication: SAMS
5. Neal Goldstein, Tony Bove, “iPhone Application Development All-In-One For Dummies”, John Wiley & Sons
6. Henry Lee, Eugene Chuvyrov, “Beginning Windows Phone App Development”, Apress, 2012.
7. Jochen Schiller, “Mobile Communications”, Addison-Wesley, 2<sup>nd</sup> edition, 2004.
8. Stojmenovic and Cacute, “Handbook of Wireless Networks and Mobile Computing”, Wiley, 2002, ISBN 0471419028.
9. Worklight resources

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

1. understand the basics of operating systems like Android, iOS for mobile devices;
2. know the architecture of mobile operating systems; and
3. understand the design and development of Application using mobile application platforms.

**BTCS-903 Ethical Hacking(Elective-I)**

<b>Internal Marks:</b>	<b>40</b>	<b>L</b>	<b>T</b>	<b>P</b>
<b>External Marks:</b>	<b>60</b>	<b>3</b>	<b>1</b>	<b>0</b>
<b>Total Marks:</b>	<b>100</b>			

**Objective:** To understand the concepts of ethical hacking and related issues

**Introduction:** Understanding the importance of security, Concept of ethical hacking and essential Terminologies-Threat, Attack, Vulnerabilities, Target of Evaluation, Exploit. Phases involved in hacking [7]

**Foot printing:** Authoritative, Non -Auth reply by DNS, Introduction to foot printing, Understanding the information gathering methodology of the hackers, Tools used for the reconnaissance phase. [7]

**Scanning:** Detecting live systems on the target network, Discovering services running /listening on target systems, Understanding port scanning techniques, Identifying TCP and UDP services running on the target network, Understanding active and passive fingerprinting. [7]

**System Hacking:** Aspect of remote password guessing, Role of eavesdropping ,Various methods of password cracking, Keystroke Loggers, Understanding Sniffers ,Comprehending Active and Passive Sniffing, ARP Spoofing and Redirection, DNS and IP Sniffing, HTTPS Sniffing. [7]

**Hacking Wireless Networks:** Introduction to 802.11,Role of WEP, Cracking WEP Keys, Sniffing Traffic, Securing Wireless Networks. [6]

**Cryptography:** Understand the use of Cryptography over the Internet through PKI, RSA, MD-5, Secure Hash Algorithm and Secure Socket Layer. [6]

**Suggested Readings/Books:**

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

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

1. Understand the basics of information security
2. Know the various concepts like foot printing, scanning and system hacking and
3. Understand hacking in wireless networks.

### **BTCS-904 Information Security (Elective-I)**

<b>Internal Marks:</b>	<b>40</b>	<b>L</b>	<b>T</b>	<b>P</b>
<b>External Marks:</b>	<b>60</b>	<b>3</b>	<b>1</b>	<b>0</b>
<b>Total Marks:</b>	<b>100</b>			

**Objectives:** Upon completion of this course, students will have gained knowledge of information security concepts and understanding of Information Security principles and approaches.

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

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

**Authentication Protocols** - Message Authentication: Authentication Requirements, Authentication Functions, Message Authentication Codes, MD5 Message Digest Algorithms, Digital Signatures and Authentication Protocols: Digital Signatures, Authentication Protocols, Digital Signature Standards. [8]

**Network Security** - Authentication Applications: Kerberos, X.509 Directory Authentication Service. Electronic Mail Security: Pretty Good Privacy. IP Security: Overview, IP Security Architecture, Authentication Header, Encapsulation Security Payload. Web Security: Web Security Requirements, Secure Sockets Layer and Transport Layer Security, Secure Electronic Transaction. [7]

**System Security-** Intruders, Malicious Software, Viruses and Related Threats, Counter Measures, Firewalls and its Design Principles. [7]

#### **Suggested Books**

1. William Stallings, Network Security Essentials, Applications and Standards Pearson Education.
2. William Stallings, Cryptography and Network Security Principles and practice. 2/e, Pearson Education.

For Batches 2015 & Onwards

Academic Autonomous Status vide letter No. F22-1/2014 (AC)

3. Bishop, Matt, Introduction to Computer Security. Addison-Wesley, Pearson Education, Inc. ISBN: 0-321-24744-2. (2005)
4. Michael. E. Whitman and Herbert J. Mattord Principles of Information Security, Cengage Learning
5. Atul Kahate Cryptography & Network Security, TMH, 2<sup>nd</sup> Edition
6. Charlie Kaufman, Radia Perlman, Mike Speciner, Network Security: Private Communication in Public World, 2nd Edition, 2011, Pearson Education.

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

1. understand the concepts of cryptography and network security;
2. know SSL and SET protocols.

BCET

**BTCS-905 Simulation and Modeling (Elective-I)**

<b>Internal Marks:</b>	<b>40</b>	<b>L</b>	<b>T</b>	<b>P</b>
<b>External Marks:</b>	<b>60</b>	<b>3</b>	<b>1</b>	<b>0</b>
<b>Total Marks:</b>	<b>100</b>			

**Objectives:** This course should provide the students with good understanding of various techniques of Simulation.

**Introduction to Simulation:** System & System Environment, Components of a System, Discrete and Continuous Systems, Model of a System and Types of Models, Discrete Event System Simulation, Advantages and Disadvantages of Simulation, Areas of Application [4]

**Techniques of Simulation:** Monte Carlo Method, Types of System Simulations, Real Time Simulation, Stochastic Variables, Discrete Probability Functions [4]

**General Principles:** Concepts in Discrete Event Simulation, Event Scheduling /Time Advance Algorithm, List Processing, Using Dynamic Allocation & Linked List [4]

**Simulation Software:** History of Simulation Software, Selection of Simulation Software, Simulation in C++, GPSS, Simulations Packages, Trends in simulation Software.[4]

**Statistical Models in Simulation:** Useful Statistical Models, Discrete Distribution s, Continuous Distributions, Poisson Process, Empirical Distributions [4]

**Queuing Models:** Characteristics of Queuing systems, Queuing Notation, Long Run Measures of performance of Queuing Systems, Steady State Behavior of infinite Population Markovian Models, Steady State Behavior of finite Population Models, Networks of Queues [5]

**Random Number Generation:** Properties of Random Numbers, Generation of Pseudo-Random Numbers, Techniques for Generating Random Numbers, Tests for Random Numbers, Inverse transform Techniques, Convolution Methods, Acceptance – Rejection Techniques [5]

**Input Modeling:** Data Collection, Identifying the Distribution with Data, Parameter Estimation, Chi – Square Test, Selecting Input Models with Data [5]

**Verification & Validation of simulation Modeling:** Model Building, Verification & Validation, Verification of simulation Models, Calibration & Validation of Models. [5]

**Suggested Books**

1. Gordon G, “*System Simulation*”, PHI 2nd Edition (1998)
2. Deo Narsingh, “*System Simulation with Digital Computers*”, PHI, New Delhi (1993)
3. K S Trivedi, “*Probability and Statistics with Reliability, Queuing and Computer Science Application*”, PHI
4. Subranranian, K R V and Sudaresan R Kadayam, “*System simulation: Introduction to GPSS*”, CBS, New Delhi (1993)
5. W Feller, ”*An introduction to Probability Theory and its Applications,*” Val 182, Wiley Eastern Ltd. ND.

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

1. Understand the concepts of system;
2. Know the various simulation techniques;
3. Know the basic and advanced features of GPSS simulation language; and
4. Understand the inventory control system

BCET



		<b>BTXXXXX</b>	<b>Open Elective-II</b>		
<b>Internal Marks:</b>	<b>40</b>		<b>L</b>	<b>T</b>	<b>P</b>
<b>External Marks:</b>	<b>60</b>		<b>3</b>	<b>0</b>	<b>0</b>
<b>Total Marks:</b>	<b>100</b>				

---

**BTCS-604 Data Mining Lab**

<b>Internal Marks:</b>	<b>30</b>	<b>L</b>	<b>T</b>	<b>P</b>
<b>External Marks:</b>	<b>20</b>	<b>0</b>	<b>0</b>	<b>2</b>
<b>Total Marks:</b>	<b>50</b>			

1. Study and usage of open source data mining tool: WEKA
2. Association rule analysis in WEKA
3. Classification using the WEKA toolkit
4. Performing data preprocessing for data mining in WEKA
5. performing clustering in WEKA
6. Introduction to exploratory data analysis using R
7. Introduction to regression using R
8. Data mining case study using the CRISP-DM standard

### **BTCS-605 Free/Open Source Software Lab**

<b>Internal Marks:</b>	<b>30</b>	<b>L</b>	<b>T</b>	<b>P</b>
<b>External Marks:</b>	<b>20</b>	<b>0</b>	<b>0</b>	<b>2</b>
<b>Total Marks:</b>	<b>50</b>			

Students will be doing the practicals related to the **Elective-I** opted by them by using open source technologies available in the area of the subject.

### **BTCS-606 Software Engineering Lab**

<b>Internal Marks:</b>	<b>30</b>	<b>L</b>	<b>T</b>	<b>P</b>
<b>External Marks:</b>	<b>20</b>	<b>0</b>	<b>0</b>	<b>2</b>
<b>Total Marks:</b>	<b>50</b>			

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

**Suggested Tools** - Visual Paradigm, Rational Software Architect. Visio, Argo UML, Rational Application Developer etc. platforms.

**BTCS-607 Java Programming Lab**

<b>Internal Marks:</b>	<b>30</b>	<b>L</b>	<b>T</b>	<b>P</b>
<b>External Marks:</b>	<b>20</b>	<b>0</b>	<b>0</b>	<b>2</b>
<b>Total Marks:</b>	<b>50</b>			

1. Implementation of classes
2. Implementation of inheritance.
3. Implementation of packages and interfaces.
4. Implementation of threads.
5. Using exception handling mechanisms.
6. Implementation of Applets.
7. Implementation of mouse events, and keyboard events.
8. Implementing basic file reading and writing methods.
9. Using basic networking features.
10. Connecting to Database using JDBC.

---

**BTCS-608 Minor Project**

<b>Internal Marks:</b>	<b>30</b>	<b>L</b>	<b>T</b>	<b>P</b>
<b>External Marks:</b>	<b>20</b>	<b>0</b>	<b>0</b>	<b>2</b>
<b>Total Marks:</b>	<b>50</b>			

Students are expected to do minor project which may be the precursor of the major project

---

---

**BTGF-600 General Fitness**

<b>Internal Marks:</b>	<b>100</b>	<b>L</b>	<b>T</b>	<b>P</b>
<b>External Marks:</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
<b>Total Marks:</b>	<b>100</b>			

---

# BCET

*Open Electives*

### BTCS-951 Data Structures(Open Elective-I)

<b>Internal Marks:</b>	<b>40</b>	<b>L</b>	<b>T</b>	<b>P</b>
<b>External Marks:</b>	<b>60</b>	<b>3</b>	<b>0</b>	<b>0</b>
<b>Total Marks:</b>	<b>100</b>			

**Objectives:** The course should provide one with a fairly good concept of the fundamentals of data structures and also of the commonly occurring algorithms. The mathematical model of data is an abstract concept of data such as set, list or graph. To make it useful for problem solving the abstraction is made concrete by going into the data structure of the model- its implementation and associated algorithms. Given a data structure, quite frequently, several alternative algorithms exist for the same operation. Naturally, the question analyzing an algorithm to determine its performance in relation to the other alternatives becomes important. The course should present the general approach towards analyzing and evaluating algorithms and while presenting an algorithm, its analysis should also be included as and when required.

**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. [4]

**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. [3]

**Linked List:** Linear linked list, operations on linear linked list, doubly linked list, operations on doubly linked list, application of linked lists. [4]

**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]

**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]

**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]

**Heaps:** Representing a heap in memory, operations on heaps, application of

heap in implementing priority queue and heap sort algorithm. [4]

**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. [4]

**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. [2]

**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. [4]

**Suggested Books:**

1. Sartaj Sahni, **Data Structures, Algorithms and Applications in C++**, Tata McGraw Hill.
2. Tenenbaum, Augenstein, & Langsam, **Data Structures using C and C++**, Prentice Hall of India.
3. R. S. Salaria, **Data Structures & Algorithms Using C++**, Khanna Book Publishing Co. (P) Ltd.
4. Seymour Lipschutz, **Data Structures**, Schaum's Outline Series, Tata McGraw Hill
5. Kruse, **Data Structures & Program Design**, Prentice Hall of India.
6. Michael T. Goodrich, Roberto Tamassia, & David Mount, **Data Structures and Algorithms in C++** (Wiley India)
7. Thomas H Cormen, Charles E Leiserson, Ronald L Rivest , and Clifford Stein, Introduction to Algorithms.
8. Ellis Horowitz, Sartaj Sahni, & Dinesh Mehta, Fundamentals of Data Structures in C++.
9. Malik , **Data Structures using C++**, Cengage Learning.

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

1. Basic ability to analyze algorithms and to determine algorithm correctness and time efficiency class.
2. Master a variety of advanced abstract data type (ADT) and data structures and their implementations.
3. Master different algorithm design techniques (divide and conquer, greedy, etc.)
4. Ability to apply and implement learned algorithm design techniques and data structures to solve problems.

## BTCS - 952 Web Technologies (Open Elective-I)

<b>Internal Marks:</b>	<b>40</b>	<b>L</b>	<b>T</b>	<b>P</b>
<b>External Marks:</b>	<b>60</b>	<b>3</b>	<b>0</b>	<b>0</b>
<b>Total Marks:</b>	<b>100</b>			

**Objective :** To understand the concept of web programming

**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. (2)

**HTML-5 and CSS:** Introduction to HTML, Lists, adding graphics to HTML page, creating tables, linking documents, forms, DHTML and Cascading Style sheets, Responsive Web Design. (7)

**Java Script:** Introduction, programming constructs: variables, operators and expressions, conditional checking, functions and dialog boxes, JavaScript DOM, creating forms, introduction to Cookies, Lightbox. (5)

**Introduction** to JQuery, JSON, Angular JS, Node.JS, Wordpress, DSN, API's. (8)

**BOOTSTRAP:** Introduction to Grid System, Topography, Tables, Images, Themes, Forms, Inputs. (4)

**XML:** Why XML, XML syntax rules, XML elements, XML attributes, XML DTD displaying XML with CSS. (2)

**AJAX:** Introduction, HTTP request, XMLHttpRequest, AJAX Server Script.(2)

**PHP:** Introduction, syntax, statements, operators, sessions, E-mail, PHP and MySQL. (10)

### **Suggested Books:**

1. HTML,CSS, JavaScript and PHP, Wiley India Textbooks.
2. Ivan Bayross: Web Enabled Commercial Application
3. Deitel,Deitel, Nieto, and Sandhu: XML How to Program, Pearson Education.
4. Online Free Resources from the Internet

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

1. Understand the basics of Internet.
2. Know the tags of HTML, Javascript etc.
3. Understand the Bootstrap, XML, AJAX and PHP Programming.

### BTCS-953 Information Security (Open Elective-I)

<b>Internal Marks:</b>	<b>40</b>	<b>L</b>	<b>T</b>	<b>P</b>
<b>External Marks:</b>	<b>60</b>	<b>3</b>	<b>0</b>	<b>0</b>
<b>Total Marks:</b>	<b>100</b>			

**Objectives:** Upon completion of this course, students will have gained knowledge of information security concepts and understanding of Information Security principles and approaches.

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

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

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

**Network Security** - Authentication Applications: Kerberos, X.509 Directory Authentication Service. Electronic Mail Security: Pretty Good Privacy. IP Security: Overview, IP Security Architecture, Authentication Header, Encapsulation Security Payload. Web Security: Web Security Requirements, Secure Sockets Layer and Transport Layer Security, Secure Electronic Transaction. [8]

**System Security**- Intruders, Malicious Software, Viruses and Related Threats, Counter Measures, Firewalls and its Design Principles. [7]

#### Suggested Books

- 1 William Stallings, Network Security Essentials, Applications and Standards Pearson Education.
2. William Stallings, Cryptography and Network Security Principles and practice. second edition ,Pearson Education.
3. Bishop, Matt, Introduction to Computer Security. Addison-Wesley, Pearson Education, Inc. ISBN: 0-321-24744-2. (2005)
4. Michael. E. Whitman and Herbert J. Mattord Principles of Information Security, Cengage Learning



5. Atul Kahate Cryptography & Network Security, TMH, 2<sup>nd</sup> Edition
6. Charlie Kaufman, Radia Perlman, Mike Speciner, Network Security: Private Communication in Public World, 2nd Edition, 2011, Pearson Education.

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

1. understand the concepts of cryptography and network security;
2. know SSL and SET protocols.

BCET

For Batches 2015 & Onwards  
Academic Autonomous Status vide letter No. F22-1/2014 (AC)  
**BTCS-961 Computer Networks–I(Open Elective-II)**

<b>Internal Marks: 40</b>	<b>L</b>	<b>T</b>	<b>P</b>
<b>External Marks: 60</b>	<b>3</b>	<b>0</b>	<b>0</b>
<b>Total Marks: 100</b>			

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

**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. [6]

**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]

**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]

**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]

**Network Layer:** Design issues, IPv4 classful and classless addressing, subnetting, Routing algorithms: distance vector and link state routing, Congestion control: Principles of Congestion Control, Congestion prevention policies, Leaky bucket and token bucket algorithms. [6]

**Transport Layer:** Elements of transport protocols: addressing, connection establishment and release, flow control and buffering, multiplexing and demultiplexing, crash recovery, introduction to TCP/UDP protocols and their comparison. [5]

**Application Layer:** World Wide Web (WWW), Domain Name System (DNS), E-mail, File Transfer Protocol (FTP), Introduction to Network security [5]

**Suggested Books:**

1. Computer Networks, 4<sup>th</sup> Edition, Pearson Education by Andrew S. Tanenbaum
2. Data Communication & Networking, 4th Edition, Tata McGraw Hill. By Behrouz A. Forouzan.

3. Computer Networking, 3<sup>rd</sup> 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.
5. Guide to Networking Essentials, 5<sup>th</sup> Edition, Cengage Learning by Greg Tomsho,
6. Handbook of Networking, Cengage Learning by Michael W. Graves.

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

1. Know the various Computer Networks
2. Understand the various transmission medias for communication;
3. Understand the encoding & decoding techniques;
4. Understand modulation and Demodulation;
5. Know multiplexing and demultiplexing techniques;
6. Understand error detection and correction techniques; and understand the protocol architecture.

BCET

For Batches 2015 & Onwards  
Academic Autonomous Status vide letter No. F22-1/2014 (AC)  
**BTCS-962 Operating Systems(Open Elective-II)**

<b>Internal Marks: 40</b>	<b>L</b>	<b>T</b>	<b>P</b>
<b>External Marks: 60</b>	<b>3</b>	<b>0</b>	<b>0</b>
<b>Total Marks: 100</b>			

**Objectives:** This course should provide the students with good understanding of Operating System including its architecture and all its components. Good conceptions on all the topics 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.

**Introduction:** Operating system, Role of Operating System as resource manager, function of kernel and shell, operating system structures, views of an operating system. [6]

**Process management:** CPU scheduling, Scheduling Algorithms, PCB, Process synchronization, Deadlocks, Prevention, Detection and Recovery [6]

**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 [9]

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

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

Brief study to multiprocessor and distributed operating systems. [5]

**Case Studies:** LINUX / UNIX Operating System and Windows based operating systems. Recent trends in operating system. [4]

**Suggested 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. 3<sup>rd</sup> Edition
4. **Operating System** by Madnick Donovan
5. **Operating System** by Stallings
6. Ida M.Flynn **Understanding Operating Systems** -, Cengage Learning

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

1. Describe, contrast and compare differing structures for operating systems
2. Understand and analyze theory and implementation of: processes, resource control (concurrency etc.), physical and virtual memory, scheduling, I/O and files
3. Study of various operating systems such as Windows, Linux, Unix etc.

BCET

For Batches 2015 & Onwards  
Academic Autonomous Status vide letter No. F22-1/2014 (AC)  
**BTCS-963 Microprocessors and Assembly Language Programming**  
**(Open Elective-II)**

<b>Internal Marks:</b>	<b>40</b>	<b>L</b>	<b>T</b>	<b>P</b>
<b>External Marks:</b>	<b>60</b>	<b>3</b>	<b>0</b>	<b>0</b>
<b>Total Marks:</b>	<b>100</b>			

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

**Introduction:** Introduction to Microprocessors, history, classification, recent microprocessors. [7]

**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. [6]

**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]

**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]

**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. [7]

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

**Suggested 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 (CO):** The expected outcomes are:

1. Students should be able to solve basic binary math operations using the microprocessor.
2. Students should be able to demonstrate programming proficiency using the various addressing modes and data transfer instructions of the target microprocessor.
3. Students should be able to program using the capabilities of the stack, the program counter, and the status register and show how these are used to execute a machine code program.
4. Students should be able to apply knowledge of the microprocessor's internal registers and operations by use of a PC based microprocessor simulator.
5. Students should be able to write assemble assembly language programs, assemble into machine a cross assembler utility and download and run their program on the training boards.

BCET

**BTCS-964 Software Engineering(Open Elective-II)**

<b>Internal Marks: 40</b>	<b>L</b>	<b>T</b>	<b>P</b>
<b>External Marks: 60</b>	<b>3</b>	<b>0</b>	<b>0</b>
<b>Total Marks: 100</b>			

**Objective:** To understand the basic and advanced concepts of software engineering.

**Evolution and impact of Software engineering,** Software life cycle models: Waterfall, prototyping, Evolutionary, and Spiral models. Feasibility study, Functional and Non-functional requirements, Requirements gathering, Requirements analysis and specification. [10]

**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 [9]

**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. [10]

**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 [11]

**Suggested Books:**

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

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

1. understand the software principles
2. understand the various phases of SDLC life cycle;
3. know the software project management;
4. understand the quality management like CMMI and Six Sigma; and
5. Understand the implementation of software in 3GL and \$GL environment.