Study Scheme & Syllabus of

Bachelor of Technology

Computer Science & Engineering

Batch 2022 onwards

(3rd -8th Semester)

By

Department of Computer Science and Engineering

Shaheed Bhagat Singh State University
Ferozepur-152004 (Punjab)

Bachelor of Technology in Computer Science & Engineering

It is a Graduate (UG) Programme of 4 years duration (8 semesters)

Courses & Examination

Semester: <u>Third Semester</u>

Course Code	Type of Course	Course Title	Hours per Week			Marks Distribution		Total Marks	Credits
			L	T	P	Internal	External		
BTCS 301-C	Professional Core Courses	Data structure & Algorithms	3	0	0	40	60	100	3
BTCS 302-C	Professional Core Courses	Object Oriented Programming	3	0	0	40	60	100	3
BTCS 303-C	Professional Core Courses	Computer Organization and Architecture	3	0	0	40	60	100	3
BTCS 304-C	Professional Core Courses	Digital Electronics	3	0	0	40	60	100	3
BTAM 304-C	Basic Science Course	Mathematics-III	3	0	0	40	60	100	3
BTCS 305C	Diploma Course 1	Multimedia Animation Technology	3	0	0	40	60	100	3
BTCS 306-C	Professional Core Courses	Digital Electronics Lab	0	0	2	30	20	50	1
BTCS 307-C	Professional Core Courses	Data structure & Algorithms Lab	0	0	3	30	20	50	1
BTCS 308-C	Professional Core Courses	Object Oriented Programming lab.	0	0	2	30	20	50	1
BTCS 309-C	Professional Core Courses	IT Workshop*	0	0	2	30	20	50	1
BTCS 310-C		Summer Workshop Training	0	0	4	30	20	0	2
Total			**18**	**0**	**13**	**390**	**460**	**850**	**24**

*Syllabus to be decided by respective institute internally. It may include latest technologies.

Semester: <u>Fourth Semester</u>

Course Code	Type of Course	Course Title	Hours per Week			Marks Distribution		Total Marks	Credits
			L	T	P	Internal	External		
BTCS 401-C	Professional Core Courses	Discrete Mathematics	3	1	0	40	60	100	4
BTCS 402-C	Professional Core Courses	Operating Systems	3	0	0	40	60	100	3
BTCS 403-C	Professional Core Courses	Design & Analysis of Algorithms	3	0	0	40	60	100	3
BTHU 901 - C	Humanities & Social Sciences including Management Courses	Personality Development	3	0	0	40	60	100	3
BTCS 404-C	Diploma Course-2	Computer Peripherals and Interfacing	3	0	0	40	60	100	3
	Open Elective	Open Elective-I	3	0	0	40	60	100	3
BTCS 405-C	Professional Core Courses	Operating Systems Lab	0	0	4	30	20	50	2
BTCS 406-C	Professional Core Courses	Design & Analysis of Algorithms Lab	0	0	4	30	20	50	2
BTCS 407-C	Professional Core Courses	Computer Peripherals and Interfacing Lab	0	0	2	30	20	50	1
Total			**18**	**1**	**10**	**330**	**420**	**750**	**24**

Students will take up summer internships of 4-6 weeks at industry or reputed organizations after the 4th sem, which will be accredited in the 5th semester.

Semester: Fifth Semester

Course Code	Type of Course	Course Title	Hours per Week			Marks Distribution		Total Marks	Credits
			L	T	P	Internal	External		
BTCS 501-C	Professional Core Courses	Database Management Systems	3	0	0	40	60	100	3
BTCS 502-C	Professional Core Courses	Software Engineering	3	0	0	40	60	100	3
BTCS 503-C	Advance Diploma Course	Computer Networks	3	0	0	40	60	100	3
BTCS XXX-C	Professional Elective	Elective-I	2	0	0	40	60	100	2
BTHU 902-C	Humanities Course	Human Resource Management	3	0	0	40	60	100	3
	Open Elective	Open Elective -II	3	0	0	40	60	100	3
BTCS 504-C	Professional Core Courses	Database Management Systems Lab	0	0	3	30	20	50	1
BTCS 505-C	Advance Diploma Course Lab	Computer Networks Lab	0	0	2	30	20	50	1
BTCS XXX-C	Professional Elective	Elective-I Lab	0	0	3	30	20	50	1
BTCS 506-C	Professional Core Courses	Software Engineering Lab	0	0	2	30	20	50	1
BTCS 507-C	Professional Training	Industrial *Training	-	-	-	60	40	100	S/US
Total			17	**0**	**10**				**21**

* 4-6 weeks industrial training undertaken after 4th semester in summer vacations.

Semester: Sixth Semester

Course Code	Type of Course	Course Title	Hours per Week			Marks Distribution		Total Marks	Credits
			L	T	P	Internal	External		
BTCS 601-C	Professional Core Courses	Compiler Design	3	0	0	40	60	100	3
BTCS 602-C	Professional Core Courses	Machine Learning	3	0	0	40	60	100	3
BTCS YYY-C	Professional Elective Courses	Elective-II	2	0	0	40	60	100	2
	Open Elective Courses	Open Elective-III	3	0	0	40	60	100	3
BTCS 603-C	Advance Diploma Course	Cloud Computing	3	0	0	40	60	100	3
BTCS 604-C	Professional Core Courses	Formal Language and Automata Theory	3	0	0	40	60	100	3
BTCS 605-C	Professional Core Courses	Compiler Design Lab	0	0	2	30	20	50	1
BTCS 606-C	Professional Core Courses	Machine Learning Lab	0	0	2	30	20	50	1
BTCS 607-C	Advance Diploma Course Lab	Cloud Computing Lab	0	0	2	30	20	50	1
BTCS YYY-C	Professional Elective Courses	Elective-II lab	0	0	2	30	20	50	1
Total			**17**	**0**	**08**				**21**

Semester: <u>Seventh Semester</u>

Course Code	Type of Course	Course Title	Hours per Week			Marks Distribution		Total Marks	Credits
			L	T	P	Internal	External		
BTCS 701-C	Professional Core Courses	Network Security and Cryptography	3	0	0	40	60	100	3
BTCS 702-C	Professional Core Courses	Data Mining and Data Warehousing	3	0	0	40	60	100	3
BTCS ZZZ-C	Professional Elective	Elective- III	3	0	0	40	60	100	3
	Open Elective Courses	Open Elective-IV	3	0	0	40	60	100	3
BTCS 703-C	Professional Core Courses	Network Security and Cryptography Lab	0	0	2	30	20	50	1
BTCS 704-C	Professional Core Courses	Data Mining and Data Warehousing Lab	0	0	2	30	20	50	1
BTCS 705-C	Project	Project	0	0	8	120	80	200	4
BTCS ZZZ-C	Professional Elective	Elective-III lab	0	0	2	30	20	50	1
Total			**17**	**0**	**12**				**19**

Semester: Eighth Semester (a)

Course Code	Course Title	Hours per Week			Marks Distribution		Total Marks	Credits
		L	T	P	Internal	External		
BTCS 802-C	IOT with Andrino, ESP Rasberry Pi	1	0	2	40	60	100	2
BTCS SSS-C	Elective – IV	2	0	0	40	60	100	2
BTCS TTT-C	Elective – V	2	0	0	40	60	100	2
BTCS UUU-C	Elective – VI	2	0	0	40	60	100	2
	Open Elective – V	3	0	0	40	60	100	3
BTCS SSS-C	Elective – IV Lab	0	0	2	30	20	50	1
BTCS TTT-C	Elective – V Lab	0	0	2	30	20	50	1
BTCS UUU-C	Elective – VI Lab	0	0	2	30	20	50	1
Total		**10**	**0**	**8**				**14**

Semester: Eighth Semester (b)

Course Code	Course Title	Marks Distribution		Total Marks	Credits
		Internal	External		
BTCS 801-C	Semester Training	300	200	500	14

LIST OF ELECTIVES

BTCS XXX-C: Elective-I

BTCS 510-C	Programming in Python
BTCS 513-C	Programming in Python Lab
BTCS 515-C	Computer Graphics
BTCS 518-C	Computer Graphics Lab
BTCS 520-C	Web Technologies
BTCS 522-C	Web Technologies Lab
BTCS 521-C	Computational Biology
BTCS 523-C	Computational Biology Lab

BTCS YYY-C: Elective-II

BTCS 614-C	Software Project Management
BTCS 615-C	Software Project Management Lab
BTCS 616-C	Data Science
BTCS 617-C	Data Science Lab
BTCS 618-C	Artificial Intelligence
BTCS 619-C	Artificial Intelligence Lab
BTCS 620-C	Mobile Application Development
BTCS 621-C	Mobile Application Development Lab

BTCS ZZZ-C: Elective-III

BTCS 705-C	Deep Learning
BTCS 706-C	Deep Learning Lab
BTCS 707-C	Distributed Databases
BTCS 708-C	Distributed databases Lab
BTCS 709-C	Computer Vision
BTCS 710-C	Computer Vision Lab
BTCS 711-C	Agile Software Development
BTCS 712-C	Agile Software Development Lab

BTCS SSS-C: Elective-IV

BTCS 810-C	Ethical Hacking
BTCS 811-C	Mobile Application Development
BTCS 812-C	Agile Software Development
BTCS 813-C	Digital Image Processing
BTCS 814-C	Ethical Hacking Lab
BTCS 815-C	Mobile Application Development Lab
BTCS 816-C	Agile Software Development Lab
BTCS 817-C	Digital Image Processing Lab

BTCS TTT-CC: Elective-V

BTCS 818-C Blockchain Technologies
BTCS 819-C Blockchain Technologies Lab
BTCS 820-C Parallel Computing
BTCS 821-C Parallel Computing Lab
BTCS 822-C Adhoc and Wireless sensor networks
BTCS 823-C Adhoc and Wireless sensor networks Lab
BTCS 824-C Quantum Computing
BTCS 825-C Quantum Computing Lab

BTCS UUU-C: Elective-VI

BTCS 826-C Analysis Visualizing and Applying Data Science using Python
BTCS 827-C Malware Analysis and Reverse Engineering
BTCS 828-C Social Network Analysis
BTCS 829-C Big Data
BTCS 830-C_ Analysis Visualizing and Applying Data Science using Python
BTCS 831-C Malware Analysis and Reverse Engineering Lab
BTCS 832-C Social Network Analysis Lab
BTCS 833-C Big Data Lab

Open electives offered by CSE department:

Offered in ODD Semester:

BTCS 901-C Data Structures & Algorithms

BTCS 902-C Object-Oriented Programming

BTCS 903-C Database Management System

BTCS 904-C Computer Networks

BTCS 905-C Network Security and Cryptography

Offered in EVEN Semester:

BTCS 906-C Operating System

BTCS 907-C Machine Learning

BTCS 908-C Cloud Computing

BTCS 909-C Data Science

Third Semester

B. Tech, Computer Science & Engg.

Course Code: BTCS301-C	Course Title: Data Structure & Algorithms	3L:0T:P	3Credits

Detailed Contents:

Module 1: Introduction

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

[6 hrs] (CO1)

Module 2: Stacks and Queues

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

[10 hrs] (CO2, CO4, CO5)

Module 3: Linked Lists

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

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

[10 hrs] (CO2, CO4, CO5)

Module 4: Sorting and Hashing

Objective and properties of different sorting algorithms: Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort; Performance and Comparison among all the methods, Hashing.

[10 hrs] (CO3)

Module 5: Graph

Basic Terminologies and Representations, Graph search and traversal algorithms and complexity analysis.

[6 hrs] (CO2, CO4)

Course Outcomes:

The student will be able to:

1. For a given algorithm student will able to analyze the algorithms to determine the time and computation complexity and justify the correctness;
2. Student will be able to handle operation like searching, insertion, deletion, traversing on various Data Structures and determine time and computational complexity;
3. Student will able to write an algorithm Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort and compare their performance in term of Space and Time complexity;
4. Students will be able to choose appropriate Data Structure as applied to specific problem definition; &

5. Demonstrate the reusability of Data Structures for implementing complex iterative problems.

Suggested Books:

1. "Classic Data Structures", Samanta and Debasis, 2nd edition, PHI publishers.
2. "Fundamentals of Data Structures", Illustrated Edition by Ellis Horowitz, SartajSahni, Computer Science Press.
3. "Data Structures with C (Schaum's Outline Series)", Seymour Lipschutz, 1st edition,McGraw Hill Education.

Reference Books:

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

Course Code: BTCS302-C	Course Title: Object-Oriented Programming	3L:0T:0P	3Credits

Pre-requisites: Programming in C

Detailed Contents:

Module 1: Introduction
Overview of C++, Sample C++ program, Different data types, operators, expressions, and statements, arrays and strings, pointers & function components, recursive functions, user - defined types, function overloading, inline functions, Classes & Objects – I: classes, Scope resolution operator, passing objects as arguments, returning objects, and object assignment.

[8 hrs] (CO1)

Module 2: Classes & Objects –II
Constructors, Destructors, friend functions, Parameterized constructors, Static data members, Functions, Arrays of objects, Pointers to objects, this pointer, and reference parameter, Dynamic allocation of objects, Copy constructors, Operator overloading using friend functions, and overloading.

[8 hrs] (CO1, CO2)

Module 3: Inheritance
Base Class, Inheritance and protected members, Protected base class inheritance, Inheriting multiple base classes, Constructors, Destructors and Inheritance, passing parameters to base class constructors, Granting access, Virtual base classes.

[8 hrs] (CO3, CO4)

Module 4: Virtual functions, Polymorphism
Virtual function, calling a Virtual function through a base class reference, Virtual attribute is inherited, Virtual functions are hierarchical, pure virtual functions, Abstract classes, Using virtual functions, Early and late binding.

[8 hrs] (CO3, CO4)

Module 5: Exception Handling

Basics of exception handling, exception handling mechanism, throwing mechanism, catching mechanism, I/O System Basics, File I/0: Exception handling fundamentals, Exception handling options. C++ stream classes, Formatted I/O, fstream and the File classes, Opening and closing a file, Reading and writing text files.

[10 hrs] (CO5)

Course Outcomes:

The student will be able to:

1. Identify classes, objects, members of a class and the relationships among them needed to solve a specific problem;
2. Demonstrate the concept of constructors and destructors. And create new definitions for some of the operators;
3. Create function templates, overload function templates;
4. Understand and demonstrate the concept of data encapsulation, inheritance, polymorphism with virtual functions; &
5. Demonstrate the concept of file operations, streams in C++ and various I/O manipulators.

Suggested Books:

1. E. Balagurusamy, Object Oriented Programming with C++, Tata McGraw Hill.

Reference Books:

1. Stanley B.Lippmann, JoseeLajoie: C++ Primer, 4th Edition, Addison Wesley, 2012.
2. Herbert Schildt: The Complete Reference C++, 4th Edition, Tata McGraw Hill, 2011.

Course Code: BTCS303-C	Course Title: Computer Organization & Architecture	3L:0T:0P	3Credits

Pre-requisites: Digital Electronics

Detailed Contents:

Module 1: Functional blocks of a computer

Basic Computer Organization and Design Instruction codes, Computer registers, Computer Instructions, Timing and Control, Instruction Execution Cycle, Memory reference instructions, CPU, memory and Input/Output subsystems, Control Unit, Design of Basic Computer, Instruction set architecture of a CPU – registers.

Register Transfer and Micro Operations Register transfer language operations, arithmetic micro-operations, logic micro-operations, shift micro-operations, arithmetic logic shift unit. Design of a complete basic computer and it's working. **[10 hrs] (CO1, CO2)**

Module 2: Introduction to CPU architecture.

CPU control unit design: Design of Control Unit Control memory, Hardwired and micro-programmed design approaches, and their comparative study. Central Processing Unit General Register Organization, Stack Organization, Instruction formats, Addressing Modes, Data transfer and manipulations, Program control, RISC and CISC architecture.

Peripheral devices and their characteristics: Input-output subsystems, I/O device interface, I/O transfers – program-controlled, interrupt-driven and DMA, I/O Processor, privileged and non-privileged instructions, software interrupts and exceptions. Programs and processes –role of interrupts in process state transitions, I/O device interfaces – SCII, USB. **[12 hrs] (CO2, CO4)**

Module 3: Pipelining & Parallel Processors

Basic concepts of pipelining, throughput and speedup, Arithmetic pipeline, Instruction, vector processors, and array processors. pipeline hazards.

Introduction to parallel processors, inter-processor communication, and synchronization. Concurrent access to memory and cache coherency. **[10 hrs] (CO5)**

Module 4: Memory Organization

Semiconductor memory technologies, Memory interleaving, Main Memory, Auxiliary memory, Concept of hierarchical memory organization, cache memory, cache size vs. block size, Virtual Memory, mapping functions, replacement algorithms, write policies. **[10 hrs] (CO3)**

Course Outcomes:

The student will be able to:

1. Understand the basic structure and functional block diagram of a digital computer;
2. Study different ways of communicating with I/O devices and standard I/O interfaces.
3. Design a memory module and analyze its operation by interfacing with the CPU;
4. Classify hardwired and microprogrammed control units; &
5. Understand the concept of pipelining and its performance metrics.

Suggested Books:

1. "Computer Organization and Architecture", Moris Mano,
2. "Computer Organization and Design: The Hardware/Software Interface", 5th Edition by David A. Patterson and John L. Hennessy, Elsevier.
3. "Computer Organization and Embedded Systems", 6th Edition by Carl Hamacher, McGraw Hill Higher Education.

Reference Books:

1. "Computer Architecture and Organization", 3rd Edition by John P. Hayes, WCB/McGraw-Hill
2. "Computer Organization and Architecture: Designing for Performance", 10th Edition by William Stallings, Pearson Education.
3. "Computer System Design and Architecture", 2nd Edition by Vincent P. Heuring and Harry F. Jordan, Pearson Education.

Course Code: BTCS304-C	Course Title: Digital Electronics	3L:0T:0P	3Credits

Detailed Contents:

Module 1:

NUMBER SYSTEMS: Binary, Octal, Decimal, Hexadecimal. Number base conversions, 1's, 2's complements, signed Binary numbers. Binary Arithmetic, Binary codes: Weighted BCD, Gray code, Excess 3 code, ASCII.

LOGIC GATES: AND, OR, NOT, NAND, NOR, Exclusive-OR and Exclusive-NOR. Implementations of Logic Functions using gates, NAND-NOR implementations.

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

Module 3:
COMBINATIONAL CIRCUITS: Design procedure – Adders, Subtractors, BCD adder, Magnitude Comparator, Multiplexer/Demultiplexer, encoder/decoder, parity checker, code converters. Implementation of combinational logic using MUX, BCD to 7 segment decoder.

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. Design of Synchronous counters: state diagram, Circuit implementation. Shift registers.

Module 4:
MEMORY DEVICES: Classification of memories, RAM organization, Write operation, Read operation, Memory cycle. ROM organization, PROM, EPROM, EEPROM, Programmable logic array, Programmable array logic, complex Programmable logic devices (CPLDS), Field Programmable Gate Array (FPGA).

A/D & D/A CONVERTORS: Analog & Digital signals. sample and hold circuit, A/D and D/A conversion techniques (Weighted type, R-2R Ladder type, Counter Type, Dual Slope type, Successive Approximation type).

COURSE OUTCOME: At the end of the course, the student will be able to:

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

Suggested Readings/ Books:

1. Morris Mano, **Digital Design**, Prentice Hall of India Pvt. Ltd
2. Donald P.Leach and Albert Paul Malvino, **Digital Principles and Applications**, 5 ed., Tata McGraw HillPublishing CompanyLimited, New Delhi, 2003.
3. R.P.Jain, **Modern Digital Electronics**, 3 ed., Tata McGraw–Hill publishing company limited, New Delhi, 2003.
4. Thomas L. Floyd, **Digital Fundamentals**, Pearson Education, Inc, New Delhi, 2003
5. Ronald J. Tocci, Neal S. Widmer, Gregory L. Moss, **Digital System - Principles and Applications**, PearsonEducation.
6. Ghosal ,**Digital Electronics**, Cengage Learning.

Course Code: BTAM304-C	Course Title: Mathematics-III	3L:0T:0P	3Credits

Detailed Contents:

Module 1:
Limit, continuity for functions with severable variables, partial derivatives, total derivative, Maxima, minima and saddle points; Method of Lagrange multipliers, Multiple Integration: double and triple integrals (Cartesian and polar), Change of order of integration in double integrals, Change of variables (Cartesian to polar), Applications of double and triple integrals to find surface area and volumes. **(12Hrs) [CO1, CO2]**

Module 2:
Sequence and series, Bolzano Weirstrass Theorem, Cauchy convergence criterion for sequence, uniform convergence, convergence of positive term series: comparison test, limit comparison test, D'Alembert's ratio test, Raabe's test, Cauchy root test, p-test, Cauchy integral test, logarithmic test, Alternating series, Leibnitz test, Power series, Taylor's series, Series for exponential, trigonometric and logarithmic functions. **(13Hrs.) [CO3]**

Module 3:
Exact, linear and Bernoulli's equations, Euler's equations, Equations not of first degree: equations solvable for p, equations solvable for y, equations solvable for x and Clairaut's type.
(12 hrs.) [CO4]

Module 4:
Second and higher order linear differential equations with constant coefficients, method of variation of parameters, Equations reducible to linear equations with constant coefficients: Cauchy and Legendre's equations. **(12 hrs.) [CO5]**

Course Outcomes: At the end of the course, the student will be able to:

1. Understand the functions of several variables that are essential in most branches of engineering;
2. Apply multiple integrals to deal with areas and volumes of various structures which are quite significant in the real world;
3. Formulate and solve engineering problems related to convergence, infinite series, power series and Taylor series;
4. Create, select and utilize the learnt techniques of first degree ordinary differential equations to model real world problems &;
5. Be acquainted with the knowledge required to solve higher-order ordinary differential equations.

Textbooks/References:

1. G.B. Thomas and R.L. Finney, Calculus and Analytic geometry, 9th Edition, Pearson, Reprint, 2002.
2. T. Veerarajan, Engineering Mathematics for first year, Tata McGraw-Hill, New Delhi, 2008.
3. N.P. Bali and Manish Goyal, A text book of Engineering Mathematics, Laxmi Publications, Reprint, 2010.
4. Erwin Kreyszig, Advanced Engineering Mathematics, 9th Edition, John Wiley & Sons, 2006.
5. W.E. Boyce and R.C. DiPrima, Elementary Differential Equations and Boundary Value Problems, 9th Edition, Wiley India, 2009.
6. E.A. Coddington, An Introduction to Ordinary Differential Equations, Prentice Hall India, 1995.

Course Code: BTCS305-C	Course Title: Multimedia and Animation Technology	3L:0T:0P	3Credits

Detailed Contents:

Introduction to Multimedia Systems (06 hrs)

What is Multimedia, History of Multimedia, Quality criteria and specifications of different capturing devices, Communication devices, Storage devices, Display devices, Elements of Multimedia and different multimedia file formats, Applications of multimedia – benefits and problems.

Multimedia Hardware and Software Essentials (08 hrs)

Classes of Multimedia Systems, Components of a Multimedia System: Quality Criteria and specifications of different Capturing Devices, Communication Devices, Storage Devices, Display Devices.

Content and Project Planning, Designing and development (08 hrs)

Planning steps and process, Concept of data compression, Text encoding, Audio encoding techniques, Types of images, Capturing images using camera/scanner, coding techniques for Moving Images, Editing , Editing of images audio, text, video and graphics, navigation and user interface designing.

Using Image Processing Tools (08 hrs)

Photo-shop workshop, image editing tools, specifying and adjusting colors, using gradient tools, selection and move tools, transforming path drawing and editing tools, using channels, layers, filters and actions

Multimedia Authoring Tools (12 hrs)

Types of Authoring programmes – Icon based, Time based, Storyboarding/scripting and object-oriented working in Macromedia Flash, exploring interface using selection pf PEN tools. Working with drawing and painting tools, applying colour viewing and manipulating time line, animating, processing, guiding layers, importing and editing sound and video clips in flash

Animation Technology (06 hrs)

Definition, History of Animation, Types of animation- 2D and 3D, Basic principles of animation, Various Terms- Animation Drawings/Cels, Rough Drawings, Clean ups, Color reference drawings, Layout, Model Sheet, Key Drawings and in Betweens, Master Background, Concept Piece, Character drawing, Story Board.

LEARNING OUTCOME

After undergoing the subject, the students will be able to:

1. Define and describe multimedia functions.
2. Identify and explain the devices, hardware and software system.
3. Operate and design in graphics.
4. Use photo-shop software for drawing and editing photos. Identify the tools to create animations,
5. Reduce the size of various file formats i.e. audio, video and text

RECOMMENDED BOOKS

1. Principles of Multimedia by Parikh, Tata McGraw Hill Education Pvt Ltd , New Delhi
2. Multimedia Technologies by Banerji,Tata McGraw Hill Education Pvt Ltd , New Delhi
3. Multimedia An Introduction by Villam Casanova and Molina; Prentice Hall of India, New Delhi
4. Photo-shop for Windows Bible by Deke Maclelland IDG Books India Pvt. Ltd., New Delhi
5. The complete animation course by Chris Patmore Pub.-Baron's Educational Series.(New York)
6. Animation Unleashed by Ellen Bessen, Michael Weise Productions,2008(U.S.A)

Course Code: BTCS306-C	**Course Title: Digital Electronics Lab**	**0L:0T:2P**	**1Credits**

List of Experiments:

1. To verify the Truth-tables of all logic gates.
2. To realize and verify the Half & full adder circuits using logic gates.
3. To realize Half & full subtractor circuits using logic gates.
4. To realize Encoder and Decoder circuits
5. To realize Multiplexer circuits
6. To realize 4-bit binary-gray & gray-binary converters.
7. To realize comparator circuit for two binary numbers of 2-bit each.
8. To realize Full adder & full subtractor circuits using encoder.
9. To design Full adder & full subtractor circuits using multiplexer.
10. To design and verify the Truth tables of all flip-flops.
11. To design Mod-6/Mod-9 synchronous up-down counter.

Course Outcomes

At the end of this course student will demonstrate the ability to:

1. Realize combinational circuits using logic gates.
2. Realize sequential circuits using logic gates.
3. Realize various types of Flip-flops and counters

Course Code: BTCS307-C	Course Title: Data Structure & Algorithms Lab	0L:0T:3P	1Credits

List of Experiments:

Task 1: Write a program to insert a new element at end as well as at a given position in an array.

Task 2: Write a program to delete an element from a given whose value is given or whose position is given.

Task 3: Write a program to find the location of a given element using Linear Search.

Task 4: Write a program to find the location of a given element using Binary Search.

Task 5: Write a program to implement push and pop operations on a stack using linear array.

Task 6: Write a program to convert an infix expression to a postfix expression using stacks.

Task 7: Write a program to evaluate a postfix expression using stacks.

Task 8: Write a recursive function for Tower of Hanoi problem.

Task 9: Write a program to implement insertion and deletion operations in a queue using linear array.

Task 10: Write a menu-driven program to perform the following insertion operations in a single-linked list:

i. Insertion at the beginning
ii. Insertion at end
iii. Insertion after a given node
iv. Traversing a linked list

Task 11: Write a menu-driven program to perform the following deletion operations in a single-linked list:

i. Deletion at the beginning
ii. Deletion at the end
iii. Deletion after a given node

Task 12: Write a program to implement push and pop operations on a stack using linked list.

Task 13: Write a program to implement push and pop operations on a queue using linked list.

Task 14: Program to sort an array of integers in ascending order using bubble sort.

Task 15: Program to sort an array of integers in ascending order using selection sort.

Task 16: Program to sort an array of integers in ascending order using insertion sort.

Task 17: Program to sort an array of integers in ascending order using quick sort.

Task 18: Program to traverse a Binary search tree in Pre-order, In-order and Post-order.

Task 19: Program to traverse graphs using BFS.

Task 20: Program to traverse graphs using DFS.

Lab Outcomes:

The student will be able to:

1. Improve practical skills in designing and implementing basic linear data structure algorithms;
2. Improve practical skills in designing and implementing Non-linear data structure algorithms;
3. Use Linear and Non-Linear data structures to solve relevant problems;
4. Choose appropriate Data Structure as applied to specific problem definition; &
5. Implement Various searching algorithms and become familiar with their design methods.

Reference Books:

1. "Data Structures with C (Schaum's Outline Series)", Seymour Lipschutz, 1st edition,McGraw Hill Education.

Course Code: BTCS308-C	Course Title: Object-Oriented Programming Lab	0L:0T:2P	1Credits

List of Experiments:

Task 1: Write a program that uses a class where the member functions are defined Inside the class.

Task 2: Write a program that uses a class where the member functions are defined outside a class.

Task 3: Write a program to demonstrate the use of static data members.

Task 4: Write a program to demonstrate the use of const data members.

Task 5: Write a program to demonstrate the use of zero argument and parameterized constructors.

Task 6: Write a program to demonstrate the use of a dynamic constructor.

Task 7: Write a program to demonstrate the use of explicit constructor.

Task 8: Write a program to demonstrate the use of initializer list.

Task 9: Write a program to demonstrate the overloading of increment and decrement operators.

Task 10: Write a program to demonstrate the overloading of memory management operators.

Task 11: Write a program to demonstrate the typecasting of a basic type to class type.

Task 12: Write a program to demonstrate the typecasting of class type to basic type.

Task 13: Write a program to demonstrate the typecasting of class type to class type.

Task 14: Write a program to demonstrate multiple inheritance.

Task 15: Write a program to demonstrate the runtime polymorphism.

Task 16: Write a program to demonstrate the exception handling.

Task 17: Write a program to demonstrate the use of class template.

Task 18: Write a program to demonstrate the reading and writing of mixed type of data.

Lab Outcomes:

The student will be able to:

1. Develop classes incorporating object-oriented techniques;
2. Design and implement object-oriented concepts of inheritance and polymorphism;
3 Illustrate and implement STL class of containers and need for exceptions to handle errors for object-oriented programs; &
4. Design and implement any real world based problem involving GUI interface using object-oriented concepts.

Reference Books:

1. Stanley B.Lippmann, Josee Lajoie: C++ Primer, 4th Edition, Addison Wesley, 2012.
2. E. Balagurusamy, Object Oriented Programming with C++, Tata McGraw Hill.

Fourth Semester

Course Code: BTCS401-C	Course Title: Discrete Mathematics	3L:0T:P	3Credits

Detailed Contents:

Module 1:

Sets, Relation and Function: Operations and Laws of Sets, Cartesian Products, Binary Relation, Partial Ordering Relation, Equivalence Relation, Image of a Set, Sum and Product of Functions, Bijective functions, Inverse and Composite Function, Size of a Set, Finite and infinite Sets, Countable and uncountable Sets, Cantor's diagonal argument and The Power Set theorem, Schroeder-Bernstein theorem.

Principles of Mathematical Induction: The Well-Ordering Principle, Recursive definition, The Division algorithm: Prime Numbers, The Greatest Common Divisor: Euclidean Algorithm, The Fundamental Theorem of Arithmetic. **(CO1, CO2)**

Module 2:

Basic counting techniques- inclusion and exclusion, pigeon-hole principle, permutation and combination. **(CO3)**

Module 3:

Propositional Logic: Syntax, Semantics, Validity and Satisfiability, Basic Connectives and Truth Tables, Logical Equivalence: The Laws of Logic, Logical Implication, Rules of Inference, The use of Quantifiers. **Proof Techniques:** Some Terminology, Proof Methods and Strategies, Forward Proof, Proof by Contradiction, Proof by Contraposition, Proof of Necessity and Sufficiency. **(CO3, CO4)**

Module 4:

Algebraic Structures and Morphism: Algebraic Structures with one Binary Operation, Semi Groups, Monoids, Groups, Congruence Relation and Quotient Structures, Free and Cyclic Monoids and Groups, Permutation Groups, Substructures, Normal Subgroups, Algebraic Structures with two Binary Operations, Rings, Integral Domain and Fields.
Boolean Algebra and Boolean Ring, Identities of Boolean Algebra, Duality, Representation of Boolean Function, Disjunctive and Conjunctive Normal Form **(CO4)**

Module 5:

Graphs and Trees: Graphs and their properties, Degree, Connectivity, Path, Cycle, Sub Graph, Isomorphism, Eulerian and Hamiltonian Walks, Graph Colouring, Colouring maps and Planar Graphs, Colouring Vertices, Colouring Edges, List Colouring, Perfect Graph, definition properties and Example, rooted trees, trees and sorting, weighted trees and prefix codes, Bi-connected component and Articulation Points, Shortest distances. **(CO5)**

Suggested books:

1. Kenneth H. Rosen, Discrete Mathematics and its Applications, Tata McGraw – Hill
2. Susanna S. Epp, Discrete Mathematics with Applications, 4th edition, Wadsworth Publishing Co. Inc.
3. C L Liu and D P Mohapatra, Elements of Discrete Mathematics A Computer Oriented Approach, 3rd Edition by, Tata McGraw – Hill.

Suggested reference books:

1. J.P. Tremblay and R. Manohar, Discrete Mathematical Structure and Its Application to Computer Science", TMG Edition, TataMcgraw-Hill
2. Norman L. Biggs, Discrete Mathematics, 2nd Edition, Oxford University Press. Schaum's Outlines Series, Seymour Lipschutz, Marc Lipson,
3. Discrete Mathematics, Tata McGraw - Hill

Course Outcomes

1. To be able to express logical sentence in terms of predicates, quantifiers, and logical connectives
2. To derive the solution for a given problem using deductive logic and prove the solution based on logical inference
3. For a given a mathematical problem, classify its algebraic structure
4. To evaluate Boolean functions and simplify expressions using the properties of Boolean algebra
5. To develop the given problem as graph networks and solve with techniques of graph theory.

Course Code: BTCS402-C	Course Title: Operating Systems	3L:0T:0P	3Credits

Detailed Contents:

Module 1: Introduction
Concept of Operating Systems, Generations of Operating systems, Types of Operating Systems, OS Services, System Calls, Structure of an OS - Layered, Monolithic, Microkernel Operating Systems, Concept of Virtual Machine. Case study on UNIX and WINDOWS Operating System.

[6 hrs] (CO1)

Module 2: Processes
Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching
Thread: Definition, Various states, Benefits of threads, Types of threads, Concept of multithreads,
Process Scheduling: Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time; Scheduling algorithms: Pre-emptive and Non-pre-emptive, FCFS, SJF, RR; Multiprocessor scheduling: Real Time scheduling: RM and EDF.

[10 hrs] (CO2, CO3)

Module 3: Inter-process Communication
Critical Section, Race Conditions, Mutual Exclusion, Hardware Solution, Strict Alternation, Peterson's Solution, The Producer\Consumer Problem, Semaphores, Event Counters, Monitors, Message Passing, Classical IPC Problems: Reader's & Writer Problem, Dinning Philosopher Problem etc.

[8 hrs] (CO2)

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

[8 hrs] (CO3)

Module 5: Memory Management
Basic concept, Logical and Physical address map, Memory allocation: Contiguous Memory allocation –Fixed and variable partition–Internal and External fragmentation and Compaction; Paging: Principle of operation – Page allocation–Hardware support for paging, Protection and sharing, Disadvantages of paging.

Virtual Memory: Basics of Virtual Memory – Hardware and control structures – Locality of reference, Page fault, Working Set, Dirty page/Dirty bit – Demand paging, Page Replacement algorithms: Optimal, First in First Out (FIFO), Second Chance (SC), Not recently used (NRU) and Least Recently used (LRU).

[10 hrs] (CO4)

Module 6: I/O Hardware
I/O devices, Device controllers, Direct memory access Principles of I/O Software: Goals of Interrupt handlers, Device drivers, Device independent I/O software, Secondary-Storage Structure: Disk structure, Disk scheduling algorithms.

File Management: Concept of File, Access methods, File types, File operation, Directory structure, File System structure, Allocation methods (contiguous, linked, indexed), Free Space Management (bit vector, linked list, grouping), directory implementation (linear list, hash table), efficiency and performance.

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

[8 hrs] (CO5, CO6)

Course Outcomes:

The student will be able to:

1. Explain basic operating system concepts such as overall architecture, system calls, user mode and kernel mode;
2. Distinguish concepts related to processes, threads, process scheduling, race conditions and critical sections;
3. Analyze and apply CPU scheduling algorithms, deadlock detection and prevention algorithms;
4. Examine and categorize various memory management techniques like caching, paging, segmentation, virtual memory, and thrashing;
5. Design and implement file management system; &
6. Appraise high-level operating systems concepts such as file systems, disk-scheduling algorithms and various file systems.

Suggested Books:

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

Reference Books:

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

Course Code: BTCS403-C	Course Title: Design and Analysis of Algorithms	3L:0T:0P	3Credits

Pre-requisites: Data Structures

Detailed Contents:

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

[8 hrs] (CO1)

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

[10 hrs] (CO1, CO2)

Module 3: Graph and Tree Algorithms

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

[10 hrs] (CO3)

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

[8 hrs] (CO5)

Module 5: Advanced Topics
Approximation algorithms, Randomized algorithms, Heuristics and their characteristics.

[6 hrs] (CO1, CO4, CO5)

Course Outcomes:

The student will be able to:

1. For a given algorithms analyze worst-case running times of algorithms based on asymptotic analysis and justify the correctness of algorithms;
2. Explain when an algorithmic design situation calls for which design paradigm (greedy/ divide and conquer/backtrack etc.);
3. Explain model for a given engineering problem, using tree or graph, and writethe corresponding algorithm to solve the problems;
4. Demonstrate the ways to analyze approximation/randomized algorithms (expected running time, probability of error); &

5. Examine the necessity for NP class based problems and explain the use of heuristic techniques.

Suggested Books:

1. Introduction to Algorithms, 4TH Edition, Thomas H Cormen, Charles E Lieserson, Ronald L Rivest and Clifford Stein, MIT Press/McGraw-Hill.
2. Data Structures and Algorithms in C++, Weiss, 4th edition, Pearson.
3. Fundamentals of Computer Algorithms – E. Horowitz, Sartaj Saini, Galgota Publications.

Reference Books

1. Algorithm Design, 1stEdition, Jon Kleinberg and ÉvaTardos, Pearson.
2. Algorithm Design: Foundations, Analysis, and Internet Examples, Second Edition, Michael T Goodrich and Roberto Tamassia, Wiley.
3. Algorithms -- A Creative Approach, 3RD Edition, UdiManber, Addison-Wesley, Reading, MA.

Course Code: BTHU901-C	Course Title: Personality Development	3L:0T:0P	3Credits

Detailed Contents:

Unit 1

Personality: Personality: Definition& Relevance, Importance of nature & nurture in Personality Development.

Understanding of Self: Self-concept, Self-identity**,** Self-image, Self-esteem, High and low Self-esteem,

Self-awareness, Techniques of Self-Awareness: Exploration through the Johari Window, Self-acceptance

and Self-Realization

Unit 2

Emotional Intelligence – Meaning, components, Importance and Relevance, Positive and Negative emotions, Healthy and Unhealthy expression of emotions, Anger: Conceptualization and Cycle.

Developing Emotional Competence: Understanding & Developing positive emotions, Positive approach towards future, Resilience during loss and challenge Developing emotional and interpersonal competence, Self-assessment, analysis and action plan

Unit 3

Significance of Positive Attitude: Building Positive Attitude, Meaning and nature of attitude, Components and Types of attitudes, Importance and relevance of attitude

Motivation: Concept, Significance - Internal and external motives - Importance of self-motivation-Factors leading to de-motivation

Self-efficacy: Meaning, Sources, and Significance

Unit 4

Stress Management: Meaning & Nature, Types of stress, Stages of stress, Causes and symptoms of stress, Consequences of stress, Strategies for stress management

Components of Excellence: Time management: Importance of time management, Techniques of time management, Assertiveness Training**,** Handling criticism and interruptions, Building independence & interdependence, Reducing resistance to change.

Text & References Books:

1. Organizational Behaviour, Davis, K. Hoover, Judhith D. Effective Small Group and Team Communication, 2002,HarcourtCollege Publishers

2. Dick, McCann&Margerison, Charles: Team Management, 1992 Edition, viva books Bates, A. P. and Julian, J.: Sociology - Understanding Social Behaviour

3. Dressler, David and Cans, Donald: The Study of Human Interaction Lapiere, Richard. T – Social Change

4. Lindzey, G. and Borgatta, E: Sociometric Measurement in the Handbook of Social Psychology, Addison – Welsley, US. Rose, G.: Oxford Textbook of Public Health, Vol.4, 1985.

5. J William Pfeiffer (ed.) Theories and Models in Applied Behavioural Science, Vol 2, Group (1996); Pfeiffer & Company

6. Smither Robert D.; The Psychology of Work and Human Performance, 1994, Harper Collins College Publishers

7. Charles S. Carver, Michael F. Scheier , Perspectives on Personality, Pearson New International Edition, 7th edition

Course Code: BTCS404-C	Course Title: Computer Peripherals and Interfacing	3L:0T:0P	3Credits

Detailed Contents:

1. **Video Display** **(08 hrs)**
   The basic principle of working of video monitors (CRT, LCD,LED), video display adapters, video modes, Video display EGA/VGA/SVGA/PCI adapters and their architecture, Overview of raster scan, vector graphic, their main difference and relative advantages, Concept of reduction and bandwidth of monitors refreshing of screen

2. **Hardware Organization of PCs** **(07 hrs)**

   Types of motherboards and their details (Form Factor, Chipset), types of processors (INTEL, AMD) and their compatibility with motherboards, serial and parallel ports, PS/2, USB Ports, Interconnection between units, connectors and cables.

3. Storage Devices **(06 hrs)**

Types of Hard Disk Drives- EIDE, SATA, SCSI, SAS External Hard Disk. Constructional features and working of hard disk drive, optical (CD, DVD, Blue Ray) disk drive and Flash Drive, Logical structure of Hard Disk and its organization, boot record.

4. Input Devices **(06 hrs)**

Detailed working principle and troubleshooting of various input devices such as keyboard, mouse, scanner. Basic principle of touch screen, light pen, digitizers. Drivers for various input devices and their role.

5. Output Devices **(06 hrs)**

Overview of printer and its classification, impact and non-impact printer, principle and working of desk Jet, dot matrix, line Printer and laser printers (Monochrome and Colour), plotter (Piezoelectric and Thermal), and modems. Software drivers for various output devices and their role.

6. Power Supplies **(06 hrs)**

Explain the working of SMPS used in computers. On-Line/Off-Line/Line-Interactive/Uninterrupted Power Supplies (UPS), basic principle of working their importance and maintenance

7. The Basic Input/Output System **(05 hrs)**

What is BIOS? Function of BIOS, software interrupts, testing and initialization, configuring the system

8. Introduction to Raspberry Pi **(04 hrs)**

LEARNING OUTCOMES

After undergoing the subject, the students will be able to:

- Identify various types of display devices/technologies.
- Describe different types and various parts of motherboard.
- Define and describe various types of processors.
- Use and describe various storage devices.
- Identify, various input-output devices and explain their working.
- Change various BIOS features.
- Assemble/maintain and troubleshoot a system.

RECOMMENDED BOOKS

1. Hardware Trouble Shooting and Maintenance by B. Govinda Rajalu, IBM PC and Clones, Tata McGraw Hill 1991
2. The waite group writing MS DOS Device, Drives byRobert, S Lai: Addison, Wesley Publishing Co. 2nd Ed. 1992.
3. Hardware and Software of Personal Computers by SK Bose; Wiley Eastern Limited, New Delhi.
4. Microprocessors and Interfacing by Hall, Douglas: McGraw Hill
5. Microprocessors and Interfacing by Uffenbeck.
6. Fundamentals of Computers by Sukhvir Singh; Khanna Publishers, New Delhi
7. Computer Peripherals for Micro Computers, Microprocessor and PC by Levis Hahensteu
8. Inside the PC (Eight Edition) by Peter Norton; Tech Media Publication, New Delhi
9. Upgrading and Preparing PC

Course Code: BTCS405-C	Course Title: Operating Systems Lab	0L:0T:4P	2Credits

List of Experiments:

Task 1: Installation Process of various operating systems.

Task 2: Implementation of CPU scheduling algorithms to find turnaround time and waiting time. a) FCFS b) SJF c) Round Robin (pre-emptive) d) Priority.

Task 3: Virtualization, Installation of Virtual Machine Software and installation of Operating System on Virtual Machine.

Task 4: 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. 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.

Task 5: 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.

Task 6: Implementation of Bankers algorithm for the purpose of deadlock avoidance.

Lab Outcomes:

The student will be able to:

1. Understand and implement basic services and functionalities of the operating system;
2. Analyze and simulate CPU Scheduling Algorithms like FCFS, Round Robin, SJF, and Priority;
3. Implement commands for files and directories;
4. Understand and implement the concepts of shell programming;
5. Simulate file allocation and organization techniques; &
6. Understand the concepts of deadlock in operating systems and implement them in multiprogramming system.

Reference Books:

1. Operating Systems: Design and Implementation, Albert S. Woodhull and Andrew S. Tanenbaum, Pearson Education.

Course Code: BTCS406-C	Course Title: Design and Analysis of Algorithms Lab	0L:0T:4P	2Credit

List of Experiments:

Task 1: Code and analyze solutions to following problem with given strategies:
i. Knap Sack using greedy approach
ii. Knap Sack using dynamic approach

Task 2: Code and analyze to find an optimal solution to matrix chain multiplication using dynamic programming.

Task 3: Code and analyze to find an optimal solution to TSP using dynamic programming.

Task 4: Implementing an application of DFS such as:
i. to find the topological sort of a directed acyclic graph ii. to find a path from source to goal in a maze.

Task 5: Implement an application of BFS such as:
i. to find connected components of an undirected graph ii. to check whether a given graph is bipartite.

Task 6: Code and analyze to find shortest paths in a graph with positive edge weights using Dijkstra's algorithm.

Task 7: Code and analyze to find shortest paths in a graph with arbitrary edge weights using Bellman-Ford algorithm.

Task 8: Code and analyze to find shortest paths in a graph with arbitrary edge weights using Flyods' algorithm.

Task 9: Code and analyze to find the minimum spanning tree in a weighted, undirected graph using Prims' algorithm

Task 10: Code and analyze to find the minimum spanning tree in a weighted, undirected graph using Kruskals' algorithm.

Task 11: Coding any real world problem or TSP algorithm using any heuristic technique.

Lab Outcomes:

The student will be able to:

1. Improve practical skills in designing and implementing complex problems with different techniques;
2. Understand comparative performance of strategies and hence choose appropriate, to apply to specific problem definition;
3. Implement Various tree and graph based algorithms and become familiar with their design methods; &
4. Design and Implement heuristics for real world problems.

Reference Books

1. Data Structures and Algorithms in C++, Weiss, 4th edition, Pearson
2. Data Structures and Algorithms using Python and C++, David M. Reed and John Zelle, 2009 edition (available as e book), Franklin Beedle& Associates.

Course Code: BTCS 407-C	Course Title: Computer Peripherals and Interfacing Lab	0L:0T:2P	1 Credits

LIST OF PRACTICALS

1) To study the construction and working of CRT, LCD, LED (coloured and black and white monitor) and its troubleshooting .
2) To Study the components and internal parts, working of hard disk and CDROM, DVD, Flash Drives
3) To study the operations and components and internal parts of Key Board, mouse and their troubleshooting
4) Study of components and internal parts and working of DMP, Inkjet printer and Laser printer and various installation of printers
5) To study the SMPS circuit and measure its various voltages. Connecting SMPS to mother- board and other devices.
6) Study the operation and maintenance of UPS.
7) Exercise on assembling a PC with peripherals and testing the same.
8) Setup and configuration of ROM BIOS
9) Visit to nearby industry

Fifth Semester

Course Code: BTCS501-C	Course Title: Database Management Systems	3L:0T:0P	3Credits

Detailed Contents:

Module 1: Database system architecture

Data Abstraction, Data Independence, Data Definition Language (DDL), Data Manipulation Language (DML). Data models: Entity-relationship model, network model, relational and object oriented Data models, integrity constraints, data manipulation operations.

[**7hrs] (CO1,2)**

Module 2: Relational query languages

Relational algebra, Tuple and domain relational calculus, SQL3, DDL and DML constructs, Open source and Commercial DBMS - MYSQL, ORACLE, DB2, SQL server. Relational database design: Domain and data dependency, Armstrong's axioms, Normal forms, Dependency preservation, Lossless design. Query processing and optimization: Evaluation of Relational algebra expressions, Query equivalence, Join strategies, Query optimization algorithms. **[10hrs] (CO2,4)**

Module 3:

Storage strategies, Indices, B-trees, hashing. **[3hrs] (CO3)**

Module 4: Transaction processing

Concurrency control, ACID property, Serializability of scheduling, Locking and timestamp-based schedulers, Multi-version and optimistic Concurrency Control schemes, Database recovery. **[6hrs] (CO3)**

Module 5: Database Security

Authentication, Authorization and access control, DAC, MAC and RBAC models, Intrusion detection, SQL injection. **[8hrs] (CO 4,5)**

Module 6: Advanced Topics

Object-oriented and object relational databases, Logical databases, Web databases, Distributed databases. **[8hrs] (CO 5)**

Course Outcomes:

At the end of study the student shall be able to:

CO1: Write relational algebra expressions for a query and optimize the developed expressions
CO2: design the databases using ER method and normalization.
CO3: Construct the SQL queries for Open source and Commercial DBMS-MYSQL, ORACLE, and DB2.
CO4: Determine the transaction atomicity, consistency, isolation, and durability.
CO5: Implement the isolation property, including locking, time stamping based on concurrency control and Serializability of scheduling.

Text Books:

1. "Database System Concepts", 6th Edition by Abraham Silberschatz, Henry F. Korth, S. Sudarshan, McGraw-Hill.

Reference :Books

1. "Principles of Database and Knowledge–Base Systems", Vol1 by J. D. Ullman, Computer Science Press.
2. "Fundamentals of Database Systems", 5th Edition by R. Elmasri and S. Navathe, Pearson Education.
3. "Foundations of Databases", Reprint by Serge Abiteboul, Richard Hull, Victor Vianu, Addison-Wesley.

Course Code: BTCS502-C	Course Title: Software Engineering	3L:0T:0P	3Credits

Detailed Contents:

Module 1:
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.

[10hrs] **(CO1)**

Module 2:
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.

[8hrs] **(CO2)**

Module 3:
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. **[10hrs]** **(CO3)**

Module 4:
Software project management, Project planning and control, cost estimation, project scheduling using PERT and GANTT charts, cost-time relations: Rayleigh-Norden results, quality management **[8hrs]** **(CO4)**

Module 5:
ISO and SEI CMMI, PSP and Six Sigma. Computer aided software engineering, software maintenance, software reuse, Component-based software development.

[6hrs] **(CO5)**

Text Books:

1. Roger Pressman, "Software Engineering: A Practitioners Approach,(6th Edition), McGraw Hill, 1997.

Reference Books:

1. Sommerville, "Software Engineering, 7th edition", Adison Wesley, 1996.
2. Watts Humphrey, "Managing software process", Pearson education, 2003.
3. James F. Peters and Witold Pedrycz, " Software Engineering – An Engineering Approach", Wiley.
4. Mouratidis and Giorgini. "Integrating Security and Software Engineering–Advances and Future", IGP. ISBN – 1-59904-148-0.
5. Pankaj Jalote, "An integrated approach to Software Engineering", Springer/Narosa.
6. Fundamentals of Software Engineering by Rajib Mall, – PHI-3rd Edition, 2009.

Course Outcomes:

At the end of the course, the student should be able to:

CO 1: Students should be able to identify the need for engineering approach to software development and various processes of requirements analysis for software engineering problems.

CO 2: Analyse various software engineering models and apply methods for design and development of software projects.

CO 3: Work with various techniques, metrics and strategies for Testing software projects.

CO 4: Identify and apply the principles, processes and main knowledge areas for Software Project Management

CO 5: Proficiently apply standards, CASE tools and techniques for engineering software projects

Course Code: BTCS 503-C	**Course Title:** Computer Networks	**3L:0T:0P**	**3Credits**

Detailed Contents:

Module 1: Data Communication Components

Representation of data and its flow Networks, Various Connection Topology, Protocols and Standards, OSI model, Transmission Media, LAN: Wired LAN, Wireless LANs, Connecting LAN and Virtual LAN, Techniques for Bandwidth utilization: Multiplexing - Frequency division, Time division and Wave division, Concepts on spread spectrum.

[8hrs] (CO1)

Module 2: Data Link Layer and Medium Access Sublayer

Error Detection and Error Correction - Fundamentals, Block coding, Hamming Distance, CRC; Flow Control and Error control protocols - Stop and Wait, Go back – N ARQ, Selective Repeat ARQ, Sliding Window, Piggybacking, Random Access, Multiple access protocols -Pure ALOHA, Slotted ALOHA, CSMA/CDCDMA/CA. **[10 hrs] (CO2)**

Module 3: Network Layer

Switching, Logical addressing – IPV4, IPV6; Address mapping – ARP, RARP, BOOTP and DHCP–Delivery, Forwarding and Unicast Routing protocols. **[8 hrs] (CO3)**

Module 4: Transport Layer

Process to Process Communication, User Datagram Protocol (UDP), Transmission Control Protocol (TCP), SCTP Congestion Control; Quality of Service, QoS improving techniques: Leaky Bucket and Token Bucket algorithm. **[8 hrs] (CO3)**

Module 5: Application Layer

Domain Name Space (DNS), DDNS, TELNET, EMAIL, File Transfer Protocol (FTP), WWW, HTTP, SNMP, Bluetooth, Firewalls, Basic concepts of Cryptography.

[8 hrs] (CO4)

Course Outcomes: The student will be able to:

CO1: Explain the functions of the different layer of the OSI Protocol;

CO2:. Describe the function of each block of wide-area networks (WANs), local area networks (LANs) and Wireless LANs (WLANs);

CO3: Develop the network programming for a given problem related TCP/IP protocol; &

CO4: Configure DNS DDNS, TELNET, EMAIL, File Transfer Protocol (FTP), WWW, HTTP, SNMP, Bluetooth, Firewalls using open source available software and tools.

Text Books:

1. Data Communication and Networking, 4th Edition, Behrouz A. Forouzan, McGraw-Hill.
2. Data and Computer Communication, 8th Edition, William Stallings, Pearson Prentice Hall India.

Reference Books:

1. Computer Networks, 8th Edition, Andrew S. Tanenbaum, Pearson New International Edition.
2. Internetworking with TCP/IP, Volume 1, 6th Edition Douglas Comer, Prentice Hall of India.
3. TCP/IP Illustrated, Volume 1, W. Richard Stevens, Addison-Wesley, United States of America.

Course Code: BTHU 501-C	Course Title: Human Resource Development	3L:0T:0P	3Credits

Detailed Contents:

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

Procurement and Placement: Need for Human Resource Planning; Process of Human Resource Planning; Methods of Recruitment; Psychological tests and interviewing; Meaning and Importance of Placement and Induction, Employment Exchanges (Compulsory Notification of vacancies) Act 1959, The Contract Labour (Regulation & Abolition) Act 1970.

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

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

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

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

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

Recommended Text Books:

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

Recommended Reference Books:

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

Course Code: BTCS504-C	Course Title: Database Management System Lab	0L:0T:3P	1Credits

List of Experiments:

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

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

Task 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.

Task 4: Set Operators, Nested Queries, Joins, Sequences.

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

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

Task 7: Stored Procedures and Exception Handling.

Task 8: Triggers and Cursor Management in PL/SQL.

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

Course Outcomes:

CO1: This practical will enable students to retrieve data from relational databases using SQL.

CO2: students will be able to implement generation of tables using datatypes

CO3: Students will be able to design and execute the various data manipulation queries.

CO4: Students will also learn to execute triggers, cursors, stored procedures etc.

Course Code: BTCS505-C	Course Title: Computer Networks Lab	0L:0T:2P	1 Credits

List of Experiments:

Task 1: To study the different types of Network cables and network topologies.
Task 2: Practically implement and test the cross-wired cable and straight through cable using clamping tool and network lab cable tester.
Task 3: Study and familiarization with various network devices.
Task 4: Familiarization with Packet Tracer Simulation tool/any other related tool.
Task 5: Study and Implementation of IP Addressing Schemes
Task 6: Creation of Simple Networking topologies using hubs and switches
Task 7: Simulation of web traffic in Packet Tracer

Task 8: Study and implementation of various router configuration commands
Task 9: Creation of Networks using routers.
Task 10: Configuring networks using the concept of subnetting
Task 11: Practical implementation of basic network command and Network configuration commands like ping, ipconfig, netstat, tracert etc. for troubleshooting network related problems.
Task 12: Configuration of networks using static and default routes.

Course Outcomes:

The students will be able to:

CO1: Know about the various networking devices, tools and also understand the implementation of network topologies;
CO2: Create various networking cables and know how to test these cables;
CO3: Create and configure networks in packet trace rtool using various network devices and topologies;
CO4: Understand IP addressing and configure networks using the subnet in;
CO5: Configure routers using various router configuration commands.

Suggested Tools **- NS2/3, Cisco packet tracer, Netsim etc.**.

Course Code: BTCS506-C	Course Title: Software Engineering Lab	0L:0T:2P	1 Credits

List of Experiments:

Task 1: Study and usage of OpenProj or similar software to draft a project plan
Task 2: Study and usage of OpenProj or similar software to track the progress of a project
Task 3: Preparation of Software Requirement Specification Document, Design Documents and Testing Phase
Task 4: related documents for some problems

Task 5: Preparation of Software Configuration Management and Risk Management related documents

Task 6: Study and usage of any Design phase CASE tool

Task 7: To perform unit testing and integration testing

Task 8: To perform various white box and black box testing techniques

Task 9: Testing of a web site

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

ELECTIVES- I

Course Code: BTCS 510-C	Course Title: Programming in Python	3L:0T:0P	3 Credits

Detailed Contents:

Module 1:

Python Basics, Objects- Python Objects, Standard Types, Other Built-in Types, Internal Types, Standard Type Operators, Standard Type Built-in Functions, Categorizing the Standard Types, Unsupported Types Numbers - Introduction to Numbers, Integers, Floating Point Real Numbers, Complex Numbers, Operators, Built-in Functions, Related Modules Sequences - Strings, Lists, and Tuples, Mapping and Set Types.

[8hrs] (CO1)

Module 2:

FILES: File Objects, File Built-in Function [open()], File Built-in Methods, File Built-in Attributes, Standard Files, Command-line Arguments, File System, File Execution, Persistent Storage Modules, Related Modules

Exceptions: Exceptions in Python, Detecting and Handling Exceptions, Context Management, *Exceptions as Strings, Raising Exceptions, Assertions, Standard Exceptions, *Creating Exceptions, Why Exceptions (Now)?, Why Exceptions at All?, Exceptions and the sys Module, Related Modules

Modules: Modules and Files, Namespaces, Importing Modules, Importing Module Attributes, Module Built-in Functions, Packages, Other Features of Modules.

[10hrs] (CO1,2)

Module 3:

Regular Expressions: Introduction, Special Symbols and Characters, Res and Python Multithreaded Programming: Introduction, Threads and Processes, Python, Threads, and the Global Interpreter Lock, Thread Module, Threading Module, Related Modules.

[8hrs] (CO 2,3)

Module 4:

GUI Programming: Introduction, Tkinter and Python Programming, Brief Tour of Other GUIs, Related Modules and Other GUIs

WEB Programming: Introduction, Wed Surfing with Python, Creating Simple Web Clients, Advanced Web Clients, CGI-Helping Servers Process Client Data, Building CGI Application Advanced CGI, Web (HTTP) Servers. **[10hrs] (CO 4,6)**

Module 5:

Database Programming: Introduction, Python Database Application Programmer's Interface (DB-API), Object Relational Managers (ORMs), Related Modules. **[6 hrs] (CO5)**

Text Books:

1. Core Python Programming, Wesley J. Chun, Second Edition, Pearson.

Course Outcomes:

The students should be able to:

CO1: Examine Python syntax and semantics and be fluent in the use of Python flow control and functions.

CO2: Demonstrate proficiency in handling Strings and File Systems.

CO3: Create, run and manipulate Python Programs using core data structures like Lists, Dictionaries and use Regular Expressions.

CO4: Interpret the concepts of Object-Oriented Programming as used in Python.

CO5: Implement exemplary applications related to Network Programming, Web Services and Databases in Python.

Course Code: BTCS 513-C	Course Title: Programming in Python Lab	0L:0T:3P	1 Credit

Prerequisites: Students should install Python.

List of Experiments:

Task 1: Write a program to demonstrate different number data types in Python.

Task 2: Write a program to perform different Arithmetic Operations on numbers in Python.

Task 3: Write a program to create, concatenate and print a string and accessing sub-string from a given string.

Task 4: Write a python script to print the current date in the following format "Sun May 29 02:26:23 IST 2017"

Task 5: Write a program to create, append, and remove lists in python.

Task 6: Write a program to demonstrate working with tuples in python.

Task 7: Write a program to demonstrate working with dictionaries in python.

Task 8: Write a python program to find largest of three numbers.

Task 9: Write a Python program to convert temperatures to and from Celsius, Fahrenheit. [Formula: c/5 = f-32/9]

Task 10: Write a Python program to construct the following pattern, using a nested for loop

```
*
* *
* * *
* * * *
* * *
* *
*
*
```

Task 11: Write a Python script that prints prime numbers less than 20.

Task 12: Write a python program to find factorial of a number using Recursion.

Task 13: Write a program that accepts the lengths of three sides of a triangle as inputs. The

program output should indicate whether or not the triangle is a right triangle (Recall from the Pythagorean Theorem that in a right triangle, the square of one side equals the sum of the squares of the other two sides).

Task 14: Write a python program to define a module to find Fibonacci Numbers and import the module to another program.

Task 15: Write a python program to define a module and import a specific function in that module to another program.

Task 16: Write a script named copyfile.py. This script should prompt the user for the names of two text files. The contents of the first file should be input and written to the second file.

Task 17: Write a program that inputs a text file. The program should print all of the unique words in the file in alphabetical order.

Task 18: Write a Python class to convert an integer to a roman numeral.

Task 19: Write a Python class to implement pow(x, n)

Task 20: Write a Python class to reverse a string word by word.

Course Code: BTCS 515-C	Course Title: Computer Graphics	3L:0T:0P	3 Credits

Detailed Contents:

Module 1:
Overview of Computer Graphics: Basics of Computer Graphics, Applications, Video Display devices, Raster–Scan displays, Random–Scan displays, Color CRT Monitors, Flat–Panel Displays; Video Controller, Display Processor, Common Graphic Input and Output devices, Graphic File Formats, Graphics Software's.

[6hrs] (CO1)

Module 2:
Output Primitives: Line Drawing, DDA, Bresenham Line Algorithm; Mid-Point Line Algorithm, Bresenham Circle Algorithm, Midpoint Circle drawing algorithms; Midpoint Ellipse Algorithm; Flood and Boundary Filling.

[6hrs] (CO1)

Module 3:
Two-Dimensional Geometric Transformation: Translation, Rotation, Scaling, Reflection, Shearing, Matrix representations; Composite transformations.

[6hrs] (CO1,2)

Module 4:
Two-Dimensional Viewing: Viewing coordinate reference frame; Window to Viewport coordinate transformation. Point Clipping, Line Clipping, text Clipping; Cohen–Sutherland and Liang–Barskey Algorithms for line clipping; Sutherland–Hodgeman algorithm for polygon clipping.

[6hrs](CO2)

Module 5:
Three Dimensional Transformations & Viewing: Translation, Rotation, Scaling, Reflection and composite transformations. Parallel and Perspective Projections, Viewing Transformation: View Plan, View Volumes and Clipping.

[6hrs] (CO2)

Module 6:
3 D Graphics and Visibility: Plane projections and its types, Vanishing points, Specification of a

3D view. 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.

[6hrs] (CO2,3)

Module 7:

Color Models: Properties of Light, Intuitive Color Concepts, concepts of chromaticity, RGB Color Model, CMY Color Model, HLS and HSV Color Models, Conversion between RGB and CMY color Models, Conversion between HSV and RGB color models, Color Selection and Applications.

[6hrs] (CO2,3)

Module 8:

Animation: Graphics Design of Animation sequences, General Computer Animation Functions Introduction to Rendering, Raytracing, Antialiasing, Fractals, Gourard and Phong shading.

[3hrs] (CO3)

Reference Books:

1. D. Hearn and M.P. Baker, Computer Graphics: C version, 2nd Edition, PHI, 2004.
2. D.F. Rogers, Mathematical Elements for Graphics, 2nd Edition., McGraw Hill, 2004.
3. J.D. Foley et al, Computer Graphics, Principles and Practices, 2nd Edition, Addison Wasley, 2004.
4. Roy A. Plastock, Gordon Kalley, Computer Graphics, Schaum's Outline Series, 1986.

Course Outcomes: The students shall be able to:

CO1: Understand about fundamentals of Graphics to enable them to design animated scenes for virtual object creations.

CO2: Make the student present the content graphically.

CO3: Work in computer aided design for content presentation for better analogy data with pictorial representation

Course Code: BTCS 518-C	Course Title: Computer Graphics Lab	0L:0T:3P	1 Credit

List of Experiments:

Task 1: WAP to draw different geometric structures using different functions.
Task 2: Implement DDA line generating algorithm.
Task 3: Implement Bresenham's line generating algorithm.
Task 4: Implement Mid-point circle line generating algorithm.
Task 5: Implementation of Bresenham's circle drawing algorithm.
Task 6: Implementation of mid-point circle generating Algorithm.
Task 7: Implementation of ellipse generating Algorithm.
Task 8: WAP of color filling the polygon using Boundary fill and Flood fill algorithm.
Task 9: To translate an object with translation parameters in X and Y directions.

Task 10: To scale an object with scaling factors along X and Y directions.
Task 11: Program of line clipping using Cohen-Sutherland algorithm.
Task 12: To perform composite transformations of an object.
Task 13: To perform the reflection of an object about major.

Course Code: BTCS 520-C	Course Title: Web Technologies	3L:0T:0P	3 Credits

Detailed Contents:

Module 1:
Introduction: History and evolution of Internet protocols, Internet addressing, Internet Service Provider (ISP), Introduction to WWW, DNS, URL, HTTP, HTTPS, SSL, Web browsers, Cookies, Web servers, Proxy servers, Web applications. Website design principles, planning the site and navigation. **[6 hrs][CO1]**

Module 2:
HTML and DHTML: Introduction to HTML and DHTML, History of HTML, Structure of HTML Document: Text Basics, Structure of HTML Document: Images, Multimedia, Links,

Audio, Video, Table and Forms, Document Layout, HTML vs. DHTML, Meta tags, Browser architecture and Website structure. Overview and features of HTML5.
[7 hrs][CO2]

Module 3:
Style Sheets: Need for CSS, Introduction to CSS, Basic syntax and structure, Types of CSS – Inline, Internal and External CSS style sheets.CSS Properties - Background images, Colors and properties,Text Formatting, Margin, Padding, Positioning etc., Overview and features of CSS3.
[7 hrs][CO3]

Module 4:
Java Script: Introduction, JavaScript's history and versions, Basic syntax, Variables, Data types, Statements, Operators, Functions, Arrays, Objects, dialog boxes, JavaScript DOM.
[7 hrs][CO4]

Module 5:
PHP and MySQL: Introduction and basic syntax of PHP, Data types, Variables,Decision and looping with examples, String, Functions, Array, Form processing, Cookies and Sessions, E-mail, PHP-MySQL: Connection to server.
[7 hrs][CO5]

Module 6:
Ajax and JSON: AJAX Introduction, AJAX Components, Handling Dynamic HTML with

Ajax, Advantages & disadvantages, HTTP request, XMLHttpRequest Server Response.

JSON– Syntax, Schema, Data types, Objects, Reading and writing JSON on client and server. Using JSON in AJAX applications. **[8 hrs][CO6]**

Students shall be able to:

CO1. Understand and apply the knowledge of web technology stack to deploy various web services.
CO2. Analyze and evaluate web technology components for formulating web related problems.
CO3. Design and develop interactive client server internet application that accommodates user specific requirements and constraint analysis.
CO4. Program latest web technologies and tools by creating dynamic pages with an understanding of functions and objects.
CO5. Apply advance concepts of web interface and database to build web projects in multidisciplinary environments.
CO6. Demonstrate the use of advance technologies in dynamic websites to provide performance efficiency and reliability for customer satisfaction.

Text Books:

1. Jeffrey C. Jackson, "Web Technologies: A Computer Science Perspective", PearsonEducation
2. Rajkamal, "Internet and Web Technology", Tata McGraw Hill
3. Ray Rischpater, "JavaScript JSON Cookbook", Packt Publishing.
4. Ivan Bayross, "Web Enabled Commercial Application Development using HTML, DHTML JavaScript, Perl, CGI", BPB Publications.
5. Peter Moulding, "PHP Black Book", Coriolis.

Course Code: BTCS 522-C	Course Title: Web Technologies Lab	0L:0T:3P	1 Credit

List of Experiments:

1. Configuration and administration Apache Web Server.
2. Develop an HTML page to demonstrate the use of basic HTML tags, Link to different HTML page and also link within a page, insertion of images and creation of tables.
3. Develop a registration form by using various form elements like input box, text area, radio buttons, check boxes etc.
4. Design an HTML page by using the concept of internal, inline, external style sheets.
5. Create an HTML file to implement the styles related to text, fonts, links using cascading style sheets

6. Create an HTML file to implement the concept of document object model using JavaScript
7. Create an HTML page including JavaScript that takes a given set of integer numbers and

shows them after sorting in descending order.

8. Write an HTML page including any required JavaScript that takes a number from one text field in the range of 0 to 999 and shows it in another text field in words. If the number is out of range, it should show "out of range" and if it is not a number, it should show "not a number" message in the result box.
9. Create a PHP file to print any text using variable.
10. Demonstrate the use of Loops and arrays in PHP
11. Create a PHP file using GET and POST methods.
12. A simple calculator web application that takes two numbers and an operator (+, -, /, * and %) from an HTML page and returns the result page with the operation performed on the operands.
13. Implement login page contains the user name and the password of the user to authenticate with Session using PHP and MySQL, also implement this with the help of PHP-Ajax.
14. A web application for implementation:
    a. The user is first served a login page which takes user's name and password. After submitting the details the server checks these values against the data from a database and takes the following decisions.
    b. If name and password matches, serves a welcome page with user's full name.
    c. If name matches and password doesn't match, then serves "password mismatch" page
    d. If name is not found in the database, serves a registration page, where user's full name is asked and on submitting the full name, it stores, the login name, password and full name in the database (hint: use session for storing the submitted login name and password)
15. Demonstrate the use of Ajax and JSON Technologies in programming examples.
16. Demonstrate the use of web site designing tools such as Joomla, WordPress.
17. Implement at least one minor project using different technologies mentioned in theory of the subject.

Course Code: BTCS521-C	**Course Title: Computational Biology**	**3L:0T:0P**	**3 Credits**

Detailed Contents:

Module 1: Introduction

Nature and scope of life science: Branches of life sciences, Characteristics of life, Levels of Organization, Origin of life, Biochemical evolution- evolution of Proteins and Nucleotide. ***Cell Biology***: The cell as basic unit of life- Prokaryotic cell and Eukaryotic cell, Cell Structure and Function- cell membrane, cell organelles, Cell Division; Mitosis & Meiosis. ***Cell Energetics***: Laws of Thermodynamics, Photosynthesis, Anaerobic & aerobic respiration, Structure and function of mitochondria, respiratory pathways: Glycolysis, Kreb's Cycle, Electron transport chain.

[10hrs] (CO)

Module 2: More about RNA and DNA

Chromosome-Genome-Genes-Databases: Bio-molecules- DNA, RNA, Protein and amino acids, Chargaff's Rules, GC content.

Central Dogma: Replication, Transcription, Translation, Post transcriptional & post translational modifications, RNA processing, RNA splicing and RNA editing. Sense/coding and anti-sense/template strands, Genetic code. Introduction to DNA and Protein sequencing.

[10hrs] (CO)

Module 3: Proteins

Proteins and Databases: Protein structure and function, Protein Primary structure, Amino acid residues, Secondary, Tertiary, Quaternary Structure of Protein, Protein sequence databases- SwissProt/ TrEMBL, PIR, Sequence motif databases -Pfam, PROSITE, Protein structure databases. **[8hrs]** **(CO)**

Module 4: Computation and Biology

Molecular computational biology: Gene prediction, sequencing genomes, similarity search, restriction mapping, ***Sequence Analysis***: Principles and its uses, Hidden Markov models for sequence analysis. Introduction of Markov Chain and Hidden Markov models. Forward backward algorithm, Viterbi and Baum-Welch algorithms,

[14hrs] **(CO)**

Course Outcomes:

The student will be able to:

CO1: Understand the basic of cell structure, divisions involved in reproduction of a cell, and its generic functionality;

CO2: Recognize the base line elements of a RNA and DNA; including fundamental behind their complex structure;

CO3: Comprehend primary structure of the protein and various related data-sets.

CO4: Demonstrate the concept of gene sequence alignment and simulate various related algorithms for the same.

Text books

1. Pevzner, P. A., Computational Molecular Biology, PHI Learning Pvt. Ltd, ISBN-978-81-203-2550-0.
2. Ghosh, Z. and Mallick, B., Bioinformatics Principles and Applications (2008) Oxford University Press ISBN 9780195692303
3. Mount, D. W., Bioinformatics – sequence and genome analysis.

Reference Books

1. Devasena, T. (2012). Cell Biology. Published by Oxford University Press.
2. Fall, C.P., Marland, E.S., Wagner, J.M., Tyson, J.J.(2002). Computational Cell Biology. Springer
3. Becker, W. M., Kleinsmith, L. J., Hardin, J., & Raasch, J. (2003). The world of the cell (Vol. 6). San Francisco: Benjamin Cummings.
4. Rastogi, S. C. (2005). Cell biology. New Age International.
5. Reece, J. B., Taylor, M. R., Simon, E. J., & Dickey, J. (2009). Biology: concepts & connections (Vol. 3, p. 2). Pearson/Benjamin Cummings.

Course Code: BTCS523-C	Course Title: Computational Biology Lab	0L:0T:3P	1 Credit

List of Experiments:

Task 1: Introduction of Bio Python, Various Packages and its Installation.

Task 2,3: Parsing sequence file formats
- Sequences and Alphabets
- Sequences act like strings
- Slicing a sequence
- Turning Seq objects into strings
- Concatenating or adding sequences
- Changing case
- Nucleotide sequences and (reverse) complements
- Transcription
- Translation

Task 4,5: Sequence annotation objects
- The SeqRecord object
- Creating a SeqRecord
  - SeqRecord objects from scratch
  - SeqRecord objects from FASTA files
  - SeqRecord objects from GenBank files
- Feature, location and position objects
  - SeqFeature objects
  - Positions and locations
  - Sequence described by a feature or location

Task 6,7,8: BLAST
- Running BLAST over the Internet
- Running BLAST locally
  - Introduction
  - Standalone NCBI BLAST+
  - Other versions of BLAST
- Parsing BLAST output
- The BLAST record class
- Dealing with PSI-BLAST
- Dealing with RPS-BLAST

BLAST and other sequence search tools
- The SearchIO object model
  - QueryResult
  - Hit
  - HSP
  - HSPFragment
- A note about standards and conventions
- Reading search output files
- Dealing with large search output files with indexing
- Writing and converting search output files

Task 9,10: Multiple Sequence Alignment objects

- Parsing or Reading Sequence Alignments
  - Single Alignments
  - Multiple Alignments
  - Ambiguous Alignments
- Writing Alignments
  - Converting between sequence alignment file formats
  - Getting your alignment objects as formatted strings
- Manipulating Alignments
  - Slicing alignments
  - Alignments as arrays

Task 11,12,13: Sequence motif analysis using Bio.motifs

- Motif objects
  - Creating a motif from instances
  - Creating a sequence logo
- Reading motifs
  - JASPAR
  - MEME
  - TRANSFAC
- Writing motifs
- Position-Weight Matrices

Quick Reference:

http://biopython.org/DIST/docs/tutorial/Tutorial.html#htoc106
https://biopython.readthedocs.io/en/latest/Tutorial/chapter_seq_objects.html

Sixth Semester

Course Code: BTCS601-C	Course Title : Compiler Design	3L:0T:0P	3Credits

Detailed Contents:

UNIT 1: Unit I Introduction to Compilers:
Structure of a compiler – Lexical Analysis – Role of Lexical Analyzer – Input Buffering – Specification of Tokens – Recognition of Tokens – Lex – Finite Automata – Regular Expressions to Automata – Minimizing DFA. [**8 hrs., CO 1]**

Unit II : Syntax Analysis:
Role of Parser – Grammars – Error Handling – Context-free grammars – Writing a grammar, Top-Down Parsing – General Strategies Recursive Descent Parser – Predictive Parser-LL(1) Parser-Shift Reduce Parser-LR Parser-LR (0) Item Construction of SLR Parsing Table - Introduction to LALR Parser – Error Handling and Recovery in Syntax Analyzer-YACC.

[8 hrs., CO 2]

Unit III : Intermediate Code Generation:
Syntax Directed Definitions, Evaluation Orders for Syntax Directed Definitions, Intermediate Languages: Syntax Tree, Three Address Code, Types and Declarations, Translation of Expressions, Type Checking. [**8 hrs., CO 3]**

Unit IV: Run-Time Environment and Code Generation:
Storage Organization, Stack Allocation Space, Access to Non-local Data on the Stack, Heap Management – Issues in Code Generation – Design of a simple Code Generator.

[6 hrs., CO 4]

Unit V: Code Optimization:
Principal Sources of Optimization – Peep-hole optimization – DAG- Optimization of Basic Blocks-Global Data Flow Analysis – Efficient Data Flow Algorithm. [**6 hrs., CO 5]**

Course Outcomes:
After undergoing this course, the students will be able to:

CO1: Build concepts on lexical analysis.

CO2: Understand strategies of syntax analysis.

CO3: Learn techniques of Intermediate code generation.

CO4: Undestand code design issues and design code generator.

CO5: Design and develop optimized codes.

Suggested Readings/ Books:

1. A.V. Aho, Monica, R.Sethi, J.D.Ullman, “Compilers, Principles, Techniques and Tools”, Second Edition, Pearson Education/Addison Wesley, 2009.Andrew W. Appel, “Modern Compiler Implementation in Java”, Second Edition, 2009.
2. J.P. Tremblay and P.G. Sorrenson, “The Theory and Practice of Compiler Writing”, McGraw Hill, 1985.

Course Code: BTCS 602-C	Course Title : Machine Learning	3L:0T:0P	3Credits

Detailed Contents:

UNIT 1: **Introduction:** Well-Posed learning problems, Basic concepts, Designing a learning system, Issues in machine learning. Types of machine learning: Learning associations, Supervised learning, Unsupervised learning and Reinforcement learning. **[4hrs] (CO1)**

UNIT 2: Data Pre-processing: Need of Data Pre-processing, Data Pre-processing Methods: Data Cleaning, Data Integration, Data Transformation, Data Reduction; Feature Scaling (Normalization and Standardization), Splitting dataset into Training and Testing set. **[4hrs] (CO2)**

UNIT 3: Regression: Need and Applications of Regression, Simple Linear Regression, Multiple Linear Regression and Polynomial Regression, Evaluating Regression Models Performance (RMSE, Mean Absolute Error, Correlation, RSquare, Accuracy with acceptable error, scatter plot, *etc*.) **[6hrs] (CO3)**

UNIT 4 Classification: Need and Applications of Classification, Logistic Regression, Decision tree, Tree induction algorithm – split algorithm based on information theory, split algorithm based on Gini index; Random forest classification, Naïve Bayes algorithm; K-Nearest Neighbours (K-NN), Support Vector Machine (SVM), Evaluating Classification Models Performance (Sensitivity, Specificity, Precision, Recall, *etc*.). **Clustering**: Need and Applications of Clustering, Partitioned methods, Hierarchical methods, Density-based methods. **[12hrs] (CO4)**

UNIT 5 Association Rules Learning: Need and Application of Association Rules Learning, Basic concepts of Association Rule Mining, Naïve algorithm, Apriori algorithm. **Artificial Neural Network:** Need and Application of Artificial Neural Network, Neural network representation and working, Activation Functions. **Genetic Algorithms:** Basic concepts, Gene Representation and Fitness Function, Selection, Recombination, Mutation and Elitism. [**14hrs] (CO5)**

Course Outcomes:

After undergoing this course, the students will be able to:

CO1: Analyse methods and theories in the field of machine learning

CO2: Analyse and extract features of complex datasets

CO3: Deploy techniques to comment for the Regression

CO4: Comprehend and apply different classification and clustering techniques

CO5: Understand the concept of Neural Networks and Genetic Algorithm

Suggested Readings/ Books:

Text Books:

1. Mitchell M., T., Machine Learning, McGraw Hill (1997) 1stEdition.
2. Alpaydin E., Introduction to Machine Learning, MIT Press (2014) 3rdEdition.
3. Vijayvargia Abhishek, Machine Learning with Python, BPB Publication (2018)

Reference Books:

1. Bishop M., C., Pattern Recognition and Machine Learning, Springer-Verlag (2011) 2ndEdition.
2. Michie D., Spiegelhalter J. D., Taylor C. C., Campbell, J., Machine Learning, Neural and Statistical Classification. Overseas Press (1994).

Course Code: BTCS 603-C	Course Title: Cloud Computing	3L:0T:0P	3Credits

Detailed Contents:

UNIT1: **Introduction:** Definition of cloud, characteristics of cloud, historical developments & challenges ahead, the vision of cloud computing, Driving factors towards cloud, Comparing grid with utility computing, cloud computing and other computing systems, types of workload patterns for the cloud, IT as a service, Applications of cloud computing.**[8hrs] (CO1)**

UNIT2: Cloud computing concepts: Introduction to virtualization techniques, Characteristics of virtualization, Pros and Cons of virtualization Technology, Hypervisors, Types of hypervisors, Multitenancy, Application programming interfaces (API), Elasticity and scalability. **[9hrs] (CO2)**

UNIT 3: Cloud service models: Cloud service models, Infrastructure as a service (IaaS) architecture- details and example, Platform as a service (PaaS) architecture- details and example, Software as a service (SaaS) architecture-- details and example, Comparison of cloud service delivery models. **[6hrs] (CO3)**

UNIT 4: Cloud deployment models: Introduction to cloud deployment models, Public clouds, Private clouds, Hybrid clouds, Community clouds, Migration paths for cloud, Selection criteria for cloud deployment. **[6hrs] (CO4)**

UNIT 5: Security in cloud computing: Understanding security risks, Principal security dangers to cloud computing, Internal security breaches, User account and service hijacking, measures to reduce cloud security breaches
Case Studies: Comparison of existing Cloud platforms /Web Services.
[6hrs] (CO5)

Course Outcomes:

After undergoing this course, the students will be able to:
CO1: Understand the core concepts of the cloud computing paradigm
CO2: Understanding importance of virtualization along with their technologies
CO3: Analyze various cloud computing service and deployment models and apply them to solve problems on the cloud.
CO4: Implementation of various security strategies for different cloud platform

Suggested Readings/ Books

1. Raj Kumar Buyya, James Broberg, Andrezei M.Goscinski, "Cloud Computing: Principles and Paradigms",Wiley 2011
2. Anthony T. Velte, Toby J. Velte and Robert Elsenpeter, "Cloud Computing: A practical Approach", McGraw Hill, 2010.
3. Barrie Sosinsky, "Cloud Computing Bible", Wiley, 2011.
4. Judith Hurwitz, Robin Bllor, Marcia Kaufman, Fern Halper, "Cloud Computing for dummies", 2009.

Reference Books

1. Rajkumar Buyya, Christian Vecchiola, S.Thamarai Selvi, "Mastering Cloud Computing" TMH 2013.
2. George Reese "Cloud Application Architectures", First Edition, O"Reilly Media 2009.
3. Dr. Kumar Saurabh "Cloud Computing" 2nd Edition, Wiley India 2012.

Course Code: BTCS604-C	**Course Title: Formal Language & Automata Theory**	**3L:1T:0P**	**3Credits**

Detailed Contents

Module 1: Introduction

Alphabet, languages and grammars, productions and derivation, Chomsky hierarchy of languages. **[3hrs] (CO1)**

Module 2: Regular languages and finite automata:

Regular expressions and languages, deterministic finite automata (DFA) and equivalence with regular expressions, nondeterministic finite automata (NFA) and equivalence with DFA, regular grammars and equivalence with finite automata, properties of regular languages, pumping lemma for regular languages, minimization of finite automata. **[8hrs] (CO2)**

Module 3: Context-free languages and pushdown automata

Context-free grammars (CFG) and languages (CFL), Chomsky and Greibach normal forms, nondeterministic pushdown automata (PDA) and equivalence with CFG, parse trees, ambiguity in CFG, pumping lemma for context-free languages, deterministic pushdown automata, closure properties of CFLs. **[8hrs] (CO3)**

Module 4: Context-sensitive languages

Context-sensitive grammars (CSG) and languages, linear bounded automata and equivalence with CSG. **[5hrs] (CO4)**

Module 5: Turing machines

The basic model for Turing machines (TM), Turing recognizable (recursively enumerable) and Turing-decidable (recursive) languages and their closure properties, variants of Turing machines, nondeterministic TMs and equivalence with deterministic TMs, unrestricted grammars and equivalence with Turing machines, TMs as enumerators. **[8hrs] (CO 5)**

Module 6: Undecidability & Intractablity:

Church-Turing thesis, universal Turing machine, the universal and diagonalization languages, reduction between languages and Rice s theorem, undecidable problems about languages.
Intractablity: Notion of tractability/feasibility. The classes NP and co-NP, their importance. Polynomial time many-one reduction. Completeness under this reduction. Cook-Levin theorem: NP-completeness of propositional satisfiability, other variants of satisfiability. NP-complete problems from other domains: graphs (clique, vertex cover, independent sets, Hamiltonian cycle), number problem (partition), set cover **[12hrs] (CO5)**

Course Outcomes: The student will be able to:

CO1: Write a formal notation for strings, languages and machines.

CO2: Design finite automata to accept a set of strings of a language.

CO3: Design context free grammars to generate strings of context free language .

CO4: Determine equivalence of languages accepted by Push Down Automata and languages generated by context free grammars

CO5: Distinguish between computability and non-computability and Decidability and undecidability.

Text Books:

1. John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, Introduction to Automata Theory, Languages, and Computation, Pearson Education Asia.

Reference Books:

1. Harry R. Lewis and Christos H. Papadimitriou, Elements of the Theory of Computation, Pearson Education Asia.
2. Dexter C. Kozen, Automata and Computability, Undergraduate Texts in Computer Science, Springer.
3. Michael Sipser, Introduction to the Theory of Computation, PWS Publishing.
4. John Martin, Introduction to Languages and The Theory of Computation, Tata McGraw Hill.

Course Code: BTCS605-C	Course Title: Compiler Design Lab	L:0;T:0; 2P	1Credit

Sr. No.	No. List of Experiments
1	Design a lexical analyser for given language and the lexical analyser should ignore redundant spaces, tabs and new lines. It should also ignore comments. Although the syntax specification states that identifiers can be arbitrarily long, you may restrict the length to some reasonable value. Simulate the same in C language.
2	Write a C program to identify whether a given line is a comment or not.
3	Write a C program to recognize strings under 'a', 'a*b+', 'abb'.
4	Write a C program to test whether a given identifier is valid or not.
5	Write a C program to simulate lexical analyzer for validating operators.
6	Implement the lexical analyzer using JLex, flex or other lexical analyzer generating tools.
7	Write a C program for implementing the functionalities of predictive parser for the mini language specified in Note 1.
8	a) Write a C program for constructing of LL (1) parsing. b) Write a C program for constructing recursive descent parsing.
9	Write a C program to implement LALR parsing.
10	a) Write a C program to implement operator precedence parsing. b) Write a C program to implement Program semantic rules to calculate the expression that takes an expression with digits, + and * and computes the value.
11	Convert the BNF rules into YACC form and write code to generate abstract syntax tree for the mini language specified in Note 1.
12	Write a C program to generate machine code from abstract syntax tree generated by the parser. The instruction set specified in Note 2 may be considered as the target code.

Course Code: BTCS606-C	Course Title: Machine Learning Lab	L:0;T:0;2 P:	1Credit

Detailed List of Tasks:

1. Implement data pre-processing
2. Deploy Simple Linear Regression
3. Simulate Multiple Linear Regression
4. Implement Decision Tree
5. Deploy Random Forest classification
6. Simulate Naïve Bayes algorithm
7. Implement K-Nearest Neighbors (K-NN), k-Means
8. Deploy Support Vector Machine, Apriori algorithm
9. Simulate Artificial Neural Network
10. Implement the Genetic Algorithm code

Suggested Tools Python/R/MATLAB

Course Code: BTCS 607-C	Course Title: Cloud Computing Lab	L:0;T:0; P:2	1 Credit

Detailed List of Tasks:

1. Install VirtualBox/VMware Workstation on different OS.
2. Install different operating systems in VMware.
3. Simulate a cloud scenario using simulator.
4. Implement scheduling algorithms.
5. To study cloud security management.
6. To study and implementation of identity management
7. Case Study - Amazon Web Services/Microsoft Azure/Google cloud services.

Suggested Tools –Matlab, Cloudsim, AWS, Azure

ELECTIVES- II

Course Code: BTCS 614-C	Course Title: Software Project Management	3L:0T:0P	3 Credits

Detailed Contents:

MODULE 1: Introduction

Project Evaluation and Planning - Activities in Software Project Management, Overview of Project Planning, Stepwise planning, contract management, Software processes and process models. **[5hrs] (CO1)**

MODULE 2: Cost Benefit Analysis

Cost Benefit Analysis, Cash Flow Forecasting, Cost-Benefit Evaluation Techniques, Risk Evaluation. Project costing, COCOMO 2, Staffing pattern, Effect of schedule compression, Putnam's equation, Capers Jones estimating rules of thumb. **[6hrs] (CO2)**

MODULE 3: Project Scheduling

Project Sequencing and Scheduling Activities, Scheduling resources, Critical path analysis, Network Planning, Risk Management, Nature and Types of Risks, Managing Risks, Hazard Identification, Hazard Analysis, Risk Planning and Control, PERT and Monte Carlo Simulation techniques. **[8hrs] (CO3)**

MODULE 4: Monitoring & Control

Monitoring and Control- Collecting Data, Visualizing Progress, Cost Monitoring, review techniques, project termination review, Earned Value analysis, Change Control, Software Configuration Management (SCM), Managing Contracts, Types of Contracts, Stages in Contract Placement, Typical Terms of a Contract, Contract Management and Acceptance. **[8hrs] (CO4)**

MODULE 5: Quality Management

Quality Management and People Management- Introduction, Understanding Behavior, Organizational Behavior, Selecting the Right Person for The Job, Motivation, The Oldman – Hackman Job Characteristics Model, Working in Groups, Organization and team structures, Decision Making, Leadership, Organizational Structures, Stress, Health and Safety. ISO and CMMI models, Testing, and Software reliability, test automation, Overview of project management tools. **[9hrs] (CO5)**

Course Outcomes:

After undergoing this course, the students will be able to:

- CO1: Explain project management in terms of the software development process
- CO2: Estimate project cost and perform cost-benefit evaluation among projects
- CO3: Apply the concepts of project scheduling and risk management.
- CO4: Explain Software configuration management and the concepts of contract management.
- CO5: Apply quality models in software projects for maintaining software quality and reliability

Suggested Readings/Books:

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

Course Code: BTCS 615-C	**Course Title: Software Project Management Lab**	**L:0;T:0; P:2**	**1 Credits**

Detailed List of Tasks:

Task 1: Introduction to MS Project

Task 2: Create a Project Plan

- Specify project name and start (or finish) date.
- Identify and define project tasks.
- Define duration for each project task.
- Define milestones in the plan
- Define dependency between tasks

Task 3: Create Project Plan contd.

- Define project calendar.
- Define project resources.
- Specify resource type and resource rates
- Assign resources against each task
- Baseline the project plan

Task 4: Execute and Monitor the Project Plan

- Update % Complete with current task status.
- Review the status of each task.
- Compare Planned vs Actual Status
- Review the status of Critical Path
- Review resources assignation status

Task 5: Generate Dashboard and Reports

- Dashboard
- Resource Reports

- Cost Reports
- Progress Reports

Suggested Tools – MS Project, Rational Team Concert

1. Course Outcomes:

After undergoing this course, the students will be able to:

CO1: Plan and manage projects.

CO2: Consolidate and communicate information about their project.

CO3: Create Gantt charts and PERT (Project Evaluation Review Technique) chart of their project

CO4: Manage resources, assignments, work allocation and generate reports to assess project status, project cost status and resource utilization.

CO5: Identify factors affecting the critical path of their project.

Course Code: BTCS618-C	Course Title : Artificial Intelligence	3L:0T:0P	3Credits

Detailed Contents:

UNIT 1: Introduction (3 Hours)
Concept of AI, history, current status, scope, agents, environments, Problem Formulations, Review
of tree and graph structures, State space representation, Search graph and Search tree.

[8hrs] (CO1)

UNIT 2: Search Algorithms
Random search, Search with closed and open list, Depth first and Breadth first search, Heuristic search, Best first search, A* algorithm, Game Search.

[9hrs] (CO2)

UNIT 3: Probabilistic Reasoning
Probability, conditional probability, Bayes Rule, Bayesian Networks- representation, construction and inference, temporal model, hidden Markov model.

[6hrs] (CO3)

UNIT 4 Markov Decision process
MDP formulation, utility theory, utility functions, value iteration, policy iteration and partially observable MDPs. **[6hrs] (CO4)**

UNIT 5 Reinforcement Learning

Passive reinforcement learning, direct utility estimation, adaptive dynamic programming, temporal difference learning, active reinforcement learning- Q learning**.**

[6hrs] (CO5)

Course Outcomes:

After undergoing this course, the students will be able to:

CO1: Build intelligent agents for search and games

CO2: Solve AI problems by learning various algorithms and strategies

CO3: Understand probability as a tool to handle uncertainity

CO4: Learning optimization and inference algorithms for model learning

CO5: Design and develop programs for an reinforcement agent to learn and act in a structured environment

Suggested Readings/ Books:

1. Stuart Russell and Peter Norvig, "Artificial Intelligence: A Modern Approach" , 3rd Edition, Prentice Hall
2. Elaine Rich and Kevin Knight, "Artificial Intelligence", Tata McGraw Hill
3. Trivedi, M.C., "A Classical Approach to Artifical Intelligence", Khanna Publishing House, Delhi.
4. Saroj Kaushik, "Artificial Intelligence", Cengage Learning India,
5. David Poole and Alan Mackworth, "Artificial Intelligence: Foundations for Computational
Agents", Cambridge University Press 2010

Course Code: BTCS 619-C	**Course Title: Artificial Intelligence Lab**	**L:0;T:0;2 P:**	**1 Credit**

Detailed List of Tasks:

1. Write a programme to conduct uninformed and informed search.
2. Write a programme to conduct game search.
3. Write a programme to construct a Bayesian network from given data.
4. Write a programme to infer from the Bayesian network.
5. Write a programme to run value and policy iteration in a grid world.
6. Write a programme to do reinforcement learning in a grid world

Course Code: BTCS620-C	Course Title: Mobile Application Development	L:3; T:0; P:0	3Credits

Details of course:

Unit-1

Introduction to Android: The Android Developing environment, Android SDK, Introduction to Open Handset Alliance, Development Framework, Application Fundamentals; Device Compatibility, System permissions, Understanding Anatomy of Android Application, Android Development Tools (**6 hrs. CO1)**

Unit-II

Getting started with Mobility: Mobility Landscape, Mobile Platforms, Mobile apps development, Android terminologies, Application Context, Activities, Services, Intents, Receiving and Broadcasting Intents, Setting up the mobile apps development environment with emulator (**6 hrs., CO1)**

Unit-III

Building block of Mobile apps: App user Interface Designing, Layout, User Interface elements, VUIs and Mobile Apps, Text to Speech Techniques, Designing the Right UI, Activity states and lifecycle, Interaction among activities (**6 hrs., CO2)**

Unit-IV

Sprucing up Mobile apps: App functionality beyond user interface- Threads, sync task, Services-states and life cycle, Notifications, Broadcast receivers, Telephony and SMS APIs Native data handling: on device file I/O, shared preferences, mobile databases such as SQLite, Working with a content provider (**8 hrs., CO 3,4)**

Unit-V

Factors in Developing Mobile Applications: Mobile Software Engineering, Frameworks and Tools, Generic UI Development, Android User Graphics and Multimedia: Performance and Multithreading, Graphics and UI Performance, Android Graphics, Mobile Agents and Peer-to-Peer Architecture, Android Multimedia **(8 hrs., CO 4,5)**

Unit-VI

Platforms and Additional Issues: Development Process, Architecture, Design, Technology Selection, Testing, Security and Hacking, Active Transactions, More on Security

(8hrs., CO 5)

Course Outcomes:

CO 1: Describe those aspects of mobile programming that make it unique from programming for other platforms,
CO 2: Critique mobile applications on their design pros and cons,
CO 3: Utilize rapid prototyping techniques to design and develop sophisticated mobile interfaces,

CO 4: Program mobile applications for the Android operating system that use basic and advanced phone features, and
CO 5: Deploy applications to the Android marketplace for distribution

References:

1. Rick Rogers, John Lombardo, Meike Blake, "Android application development", Ist Edition, O'Reilly, 2010

2. Lauren Darcey and Shane Conder,"Android Wireless Application Development", 2nd ed. Pearson Education, 2011

3. Wei-Meng Lee , Beginning Android 4 development ,2012 by John Wiley & Sons

4. Jeff Mewherter, Scott Gowell,Wrox Publisher,"Professional Mobile Application Development", Ist Edition, 2012

5. Reto Meier, "Professional Android 4 Application Development", Wrox, 2012

Course Code: BTCS621-C	**Course Title: Mobile Application Development Lab**	**L:0; T:0;P:2**	**1Credit**

LIST OF PRACTICALS

1. Introduction to Android platform. Introduction to the tools used in the lab. Create a simple application
2. Understand the app idea and design user interface/wireframes of mobile app
3. Set up mobile app development environment
4. Write a program using activity class to show different events.
5. Write a program to convert text to speech.
6. Develop and debug mobile app components – User interface, services, notifications, broadcast receivers, data components
7. Using emulator to deploy and run mobile apps
8. Testing mobile app- unit testing, black box testing and test automation

SEVENTH SEMESTER

Course Code: BTCS 701-C	Course Title: Network Security and Cryptography	3L:0T:0P	3Credits

Detailed Contents:

UNIT 1: Introduction

Introduction to Cryptography, Security Threats, Vulnerability, Active and Passive attacks, Security services and mechanism, Conventional Encryption Model, CIA model

[5hrs]

(CO1)

UNIT 2: Math Background

Modular Arithmetic, Euclidean and Extended Euclidean algorithm, Prime numbers, Fermat and Euler's Theorem **[5hrs](CO1)**

UNIT 3: Cryptography

Dimensions of Cryptography, Classical Cryptographic Techniques Block Ciphers (DES, AES) : Feistal Cipher Structure, Simplifies DES, DES, Double and Triple DES, Block Cipher design Principles, AES, Modes of Operations Public-Key Cryptography : Principles Of Public-Key Cryptography, RSA Algorithm, Key Management, Diffie-Hellman Key Exchange,
Elgamal Algorithm, Elliptic Curve Cryptography. **[12hrs] (CO2)**

UNIT 4 Hash and MAC Algorithms

Authentication Requirement, Functions, Message Authentication Code, Hash Functions, Security Of Hash Functions And Macs, MD5 Message Digest Algorithm, Secure Hash Algorithm, Digital Signatures, Key Management : Key Distribution Techniques, Kerberos

[6hrs] (CO3)

UNIT 5 Security in Networks

Threats in networks, Network Security Controls – Architecture, Encryption, Content Integrity, Strong Authentication, Access Controls, Wireless Security, Honeypots, Traffic flow security, Firewalls – Design and Types of Firewalls, Personal Firewalls, IDS, Email Security – PGP,
S/MIME. **[7hrs] (CO4)**

Course Outcomes:

After undergoing this course, the students will be able to:

CO1: Understand the fundamental principles of access control models and techniques, authentication and secure system design

CO2: Have a strong understanding of different cryptographic protocols and techniques and be able to use them.

CO3: Apply methods for authentication, access control, intrusion detection and prevention. CO4: Identify and mitigate software security vulnerabilities in existing systems.

Suggested Readings/ Books:

1. Cryptography And Network Security Principles And Practice Fourth Edition, William Stallings, Pearson Education

2. Modern Cryptography: Theory and Practice, by Wenbo Mao, Prentice Hall PTR

3. Network Security Essentials: Applications and Standards, by William Stallings. Prentice Hall

4. Cryptography: Theory and Practice by Douglas R. Stinson, CRC press.

Course Code: BTCS702-C	Course Title: Data Warehousing and Data Mining	3L: 0T: 0P	Credits: 3

Detailed Contents:

UNIT 1:

Data Warehousing Introduction: design guidelines for data warehouse implementation, Multidimensional Models; OLAP- introduction, Characteristics, Architecture, Multidimensional view Efficient processing of OLAP Queries, OLAP server Architecture ROLAP versus MOLAP Versus HOLAP and data cube, Data cube operations, data cube computation.

Data mining: What is data mining, Challenges, Data Mining Tasks, Data: Types of Data, Data Quality, Data Pre-processing, Measures of Similarity and Dissimilarity **[10hrs]**

UNIT 2:

Data mining: Introduction, association rules mining, Naive algorithm, Apriori algorithm, direct hashing and pruning (DHP), Dynamic Item set counting (DIC), Mining frequent pattern without candidate generation (FP, growth), performance evaluation of algorithms

Classification: Introduction, decision tree, tree induction algorithms – split algorithm based on information theory, split algorithm based on Gini index; naïve Bayes method; estimating predictive accuracy of classification method **[10 hrs]**

UNIT 3:

Cluster analysis: Introduction, partition methods, hierarchical methods, density based methods, dealing with large databases, cluster software

Search engines: Characteristics of Search engines, Search Engine Functionality, Search Engine Architecture, Ranking of web pages, The search engine history, Enterprise Search, Enterprise Search Engine Software. **[10 hrs]**

UNIT 4:

Web data mining: Web Terminology and Characteristics, Locality and Hierarchy in the web, Web Content Mining, Web Usage Mining, Web Structure Mining, Web mining Software.**[8 hrs]**

Suggested Readings / Books:

1. Carlo Vercellis, Business Intelligence: Data mining and Optimization for Decision Making, WILEY.
2. Han J., Kamber M. and Pei J. , b Data mining concepts and techniques, Morgan Kaufmann Publishers (2011) 3rd ed.
3. Pudi V., Krishana P.R., Data Mining, Oxford University press, (2009) 1st ed.
4. Adriaans P., Zantinge D., Data mining, Pearsoneducation press (1996), 1st ed.
5. Pooniah P. , Data Warehousing Fundamentals, Willey interscience Publication, (2001), 1st ed.

ELECTIVES- III

Course Code: BTCS 705-C	Course Title : Deep Learning	3L:0T:0P	3Credits

Detailed Contents:

UNIT 1: **Machine Learning Basics:** Learning, Under-fitting, Overfitting, Estimators, Bias, Variance, Maximum Likelihood Estimation, Bayesian Statistics, Supervised Learning, Unsupervised Learning and Stochastic Gradient Decent. **[4hrs] (CO1)**

UNIT 2: Deep Feedforward Network: Feed-forward Networks, Gradient-based Learning, Hidden Units, Architecture Design, Computational Graphs, Back-Propagation, Regularization, Parameter Penalties, Data Augmentation, Multi-task Learning, Bagging, Dropout and Adversarial Training and Optimization. **[4hrs] (CO2)**

UNIT 3: Convolution Networks: Convolution Operation, Pooling, Basic Convolution Function, Convolution Algorithm, Unsupervised Features and Neuroscientific for convolution Network. **[6hrs] (CO3)**

UNIT 4: Sequence Modelling: Recurrent Neural Networks (RNNs), Bidirectional RNNs, Encoder- Decoder Sequence-to-Sequence Architectures, Deep Recurrent Network, Recursive Neural Networks and Echo State networks. **[12hrs] (CO4)**

UNIT 5: Deep Generative Models: Boltzmann Machines, Restricted Boltzmann Machines, Deep Belief Networks, Deep Boltzmann Machines, Sigmoid Belief Networks, Directed Generative Net, Drawing Samples from Auto –encoders. **[14hrs] (CO5)**

Course Outcomes:

After undergoing this course, the students will be able to:

CO1: Comprehend the advancements in learning techniques

CO2: Compare and explain various deep learning architectures and algorithms.

CO3: Demonstrate the applications of Convolution Networks

CO4: Apply Recurrent Network for Sequence Modelling

CO5: Deploy the Deep Generative Models

Suggested Readings/ Books:

Text Books:

1. Goodfellow L., Bengio Y. and Courville A., Deep Learning, MIT Press (2016).
2. Patterson J. and Gibson A., Deep Learning: A Practitioner's Approach, O'Reilly (2017), 1st ed.

Reference Books:

1. Haykin S., Neural Network and Machine Learning, Prentice Hall Pearson (2009), 3rd ed.
2. Geron A., Hands-on Machine Learning with Sci-kit and TensorFlow, O'Reilly Media (2017)

Course Code: BTCS706-C	**Course Title: Deep Learning Lab**	**L:0;T:0; 2P:**	**Credits;1**

Detailed List of Tasks:

1. Creating a basic network and analyze its performance
2. Deploy the Confusion matrix and simulate for Overfitting
3. Visualizing a neural network
4. Demo: Object Detection with pre-trained RetinaNet with Keras
5. Neural Recommender Systems with Explicit Feedback
6. Backpropagation in Neural Networks using Numpy
7. Neural Recommender Systems with Implicit Feedback and the Triplet Loss
8. Fully Convolutional Neural Networks
9. ConvNets for Classification and Localization
10. Text Classification and Word Vectors
11. Character Level Language Model (GPU required)

Suggested Tools Python/R/MATLAB

Course Code: BTCS707-C	**Course Title:** Distributed Databases	**3L: 0T: 0P**	**Credits: 3**

Detailed Contents:

Unit 1:

INTRODUCTION: Distributed data processing; What is a DDBS; Advantages and disadvantages of DDBS; Problem areas; Overview of database and computer network concepts

DISTRIBUTED DATABASE MANAGEMENT SYSTEM ARCHITECTURE:

Transparencies in a distributed DBMS; Distributed DBMS architecture; Global directory issues.

6 hrs., CO1

Unit 2:

DISTRIBUTED DATABASE DESIGN: Alternative design strategies; Distributed design issues; Fragmentation; Data allocation.

SEMANTICS DATA CONTROL: View management; Data security; Semantic Integrity Control.

QUERY PROCESSING ISSUES: Objectives of query processing; Characterization of query processors; Layers of query processing; Query decomposition; Localization of distributed data.

2 hrs., CO1

Unit 3:

DISTRIBUTED QUERY OPTIMIZATION: Factors governing query optimization; Centralized query optimization; Ordering of fragment queries; Distributed query optimization algorithms.

TRANSACTION MANAGEMENT: The transaction concept; Goals of transaction management; Characteristics of transactions; Taxonomy of transaction models.

CONCURRENCY CONTROL: Concurrency control in centralized database systems; Concurrency control in DDBSs; Distributed concurrency control algorithms; Deadlock management. **10hrs., CO2**

Unit 4:
RELIABILITY: Reliability issues in DDBSs; Types of failures; Reliability techniques; Commit protocols; Recovery protocols.

PARALLEL DATABASE SYSTEMS : Parallel architectures; parallel query processing and optimization; load balancing.

ADVANCED TOPICS: Databases, Distributed Object Management, Multi-databases.

10 hrs., CO2,3

COURSE OUTCOMES

After completion of course, students would be able to:

CO1: Design trends in distributed systems.

CO2: Apply network virtualization in distributed

environment. CO3: Apply remote method invocation

and objects.

References:

1. Principles of Distributed Database Systems, M.T. Ozsu and P. Valduriez, Prentice-Hall, 1991.
2. Distributed Database Systems, D. Bell and J. Grimson, Addison-Wesley, 1992.

Course Code: BTCS708-C	Course Title: Distributed Databases Lab	L: T: 2P	Credits: 1

Detailed list of Tasks:

Programs may be implemented using any open source tool

Expt. 1: Installation and configuration of database packages.

Expt. 2: Creating and managing database objects (Tables, views,

indexes etc.) **Expt. 3:** Creating and managing database security

through user management. **Expt. 4:** Creating and maintaining database

links.

Expt. 5: Implement Partitioning on the database tables.

Expt. 6: Implement various Transaction concurrency control methods [i.e. lock's] by executing multiple update and queries.

Expt. 7: Performance tuning of SQL queries.

Mini Project: Student has to do a project assigned from course contents in a group of two or three students. The team will have to demonstrate as well as have to give a presentation of the same.

Course Code: BTCS709-C	Course Title: Computer Vision	3L: 0T: 0P	Credits: 3

Detailed Contents:

Unit 1: Overview, computer imaging systems, lenses, Image formation and sensing, Image analysis, pre- processing and Binary image analysis. **6 hrs., CO1**

Unit 2:

Edge detection, Edge detection performance, Hough transform, corner detection. **4 hrs., CO1**

Unit 3:

Segmentation, Morphological filtering, Fourier transform. **4 hrs., CO1**

Unit 4:

Feature extraction, shape, histogram, color, spectral, texture, using CVIP tools, Feature analysis, feature vectors, distance /similarity measures, data pre- processing.

8 hrs., CO2

Unit 5:

Pattern Analysis: Clustering: K-Means, K-Medoids, Mixture of Gaussians Classification: Discriminant Function, Supervised, Un-supervised, Semi- supervised. Classifiers: Bayes, KNN, ANN models;

Dimensionality Reduction: PCA, LDA, ICA, and Non- parametric methods.

Recent trends in Activity Recognition, computational photography, Biometrics **8 hrs.,**

CO1 COURSE OUTCOMES

After completion of course, students would be able to:

CO1: Understand image detection and analysis

CO2: Identify features to recognize object , scene and categorization from images.

CO3: Develop the skills necessary to build computer vision applications.

References:

1. Computer Vision: Algorithms and Applications by Richard Szeliski.
2. Deep Learning, by Goodfellow, Bengio, and Courville.
3. Dictionary of Computer Vision and ImageProcessing, by Fisheretal.

Course Code: BTCS710-C	Course Title: Computer Vision Lab	L: T: 2P	Credits: 1

Detailed list of Tasks:

Programs may be implemented using MATLAB/C/C++/Java/Python on binary/grayscale/color images.

Expt. 1: Implementation of basic image transformations:

a. Log
b. Power law
c. Negation

Expt. 2: Implementation the following:

a. Histogram processing
b. Histogram equalization/matching

Expt. 3: Implementation of piecewise linear transformations

a. Contrast stretching
b. Grey level slicing
c. Bit plane slicing

Expt. 4: Implementation of image enhancement/smoothing using

a. Linear (weighted and non-weighted filters)
b. Order statistics filters (Nonlinear filters)
    i. Mean
    ii. Median
    iii. Min
    iv. Max
    v. Average

Expt. 5: Implementation of image enhancement/sharpening using

a. Laplacian operators
b. Sobel's operators
c. Robert's cross operators

Expt. 6: Implement the 2D-DFT to obtain Fourier coefficients and reconstruct the image, i.e., IDFT.

Expt. 7: Implement image enhancement using Fourier low pass filters

a. deal
b. Butterworth
c. Gaussian

Expt. 8: Implement image enhancement using Fourier high pass filters

a. Ideal

b. Butterworth

c. Gaussian

Expt. 9: Implement algorithms to detect the following in an image

a. Point

b. Line

c. Boundary

Expt. 10: Implement Hough transform to detect a line.

Mini Project: Student has to do a project assigned from course contents in a group of two or three students. The team will have to demonstrate as well as have to give a presentation of the same.

Course Code: BTCS 711-C	**Course Title : Agile Software Development**	**3L:0T:0P**	**3Credits**

Detailed Contents:

UNIT 1: **Introduction**

Need of Agile software development, History of Agile, Agile context– manifesto, principles, methods, values. The benefits of agile in software development. **[6hrs] (CO1)**

UNIT 2: Agile Design Methodologies

Fundamentals, Design principles–Single responsibility, Open-closed, Liskov-substitution, Dependency-inversion, Interface-segregation. **[6hrs] (CO2)**

UNIT 3: Scrum

Introduction to scrum framework, Roles: Product owner, team members and scrum master, Events: Sprint, sprint planning, daily scrum, sprint review, and sprint retrospective, Artifacts: Product backlog, sprint backlog and increments. User stories- characteristics and contents. **[8hrs] (CO3)**

UNIT 4: Kanban

Introduction to Kanban framework, Workflow, Limit the amount of work in progress, pulling work from column to column, Kanban board, Adding policies to the board, Cards and their optimization. **[6hrs]** (**CO4)**

UNIT 5: Extreme Programming

Basic values and principles, Roles, Twelve practices of XP, Pair programming, XP team, Life cycle and tools for XP. **[6hrs] (CO5)**

UNIT 6: Agile Testing

The Agile lifecycle and its impact on testing, Test driven development– Acceptance tests and verifying stories, writing a user acceptance test, Developing effective test suites, Continuous integration, Code refactoring. Risk based testing, Regression tests, Test automation.

[6hrs] (CO6)

Course Outcomes:

After undergoing this course, the students will be able to:

CO1: Understand concept of agile software engineering and its advantages in software development.

CO2 Explain the role of design principles in agile software design.

CO3 Define the core practices behind Scrum framework.

CO4 Understand key principles of agile software development methodology-Kanban.

CO5 Describe implications of functional testing, unit testing, and continuous integration.

CO6 Understand the various tools available to agile teams to test the project.

Suggested Readings/ Books:

1. Ken Schawber, Mike Beedle, “Agile Software Development with Scrum”, Pearson.
2. Robert C. Martin, “Agile Software Development, Principles, Patterns and

Practices", Prentice Hall.
3. Mike Cohn, "User Stories Applied: For Agile Software Development", Addison Wesley Signature Series.
4. Lisa Crispin, Janet Gregory, "Agile Testing: A Practical Guide for Testers and Agile Teams", Addison Wesley.
5. Paul VII, "Agile: The Complete Overview of Agile Principles and Practices (Agile Product Management)".
6. Robert Martin, "Agile Software Development, Principles, Patterns, and Practices", Pearson New International Edition.
7. Greene Jennifer," Learning Agile", O'Reilly Series.

Course Code: BTCS 712-C	**Course Title : Agile Software Development Lab**	**L:0T:2P**	**Credits:1**

Detailed List of Tasks:

1. Understand the background and driving forces for taking an Agile Approach to Software Development.
2. Build out a backlog and user stories.
3. To study and use automated build tool.
4. To study-- version control tool.
5. To study Continuous Integration tool.
6. Apply Design principle and Refactoring to achieve agility.
7. Perform Testing activities within an agile project.
8. Mini Project: based on tools