Scheme of

Bachelor of Technology

Computer Science & Engineering

Batch 2018

(3rd - 8th Semester)

By

Department of Academics

IK Gujral Punjab Technical University

Bachelor of Technology in Computer Science & Engineering

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

Courses & Examination

Third Semester

Course Code	Type of Course	Course Title	Hours per Week			Marks Distribution		Total Marks	Credits
			L	T	P	Internal	External		
BTES 301-18	Engineering Science Course	Digital Electronics	3	0	0	40	60	100	3
BTCS 301-18	Professional Core Courses	Data structure & Algorithms	3	0	0	40	60	100	3
BTCS 302-18	Professional Core Courses	Object Oriented Programming	3	0	0	40	60	100	3
BTAM 304-18	Basic Science Course	Mathematics-III	3	0	0	40	60	100	3
HSMC 101/102-18	Humanities & Social Sciences Including Management \Courses	Foundation Course in Humanities (Development of Societies/Philosophy)	2	1	0	40	60	100	3
BTES 302-18	Engineering Science Course	Digital Electronics Lab	0	0	2	30	20	50	1
BTCS 303-18	Professional Core Courses	Data structure & Algorithms Lab	0	0	4	30	20	50	2
BTCS 304-18	Professional Core Courses	Object Oriented Programming lab.	0	0	4	30	20	50	2
BTCS 305-18	Professional Core Courses	IT Workshop*	0	0	2	30	20	50	1
		Summer Institutional Training	0	0	0	0	0	0	Satisfactory/Unsatisfactory
Total			**14**	**1**	**12**	**320**	**380**	**700**	**21**

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

Fourth Semester

Course Code	Type of Course	Course Title	Hours per Week			Marks Distribution		Total Marks	Credits
			L	T	P	Internal	External		
BTCS 401-18	Professional Core Courses	Discrete Mathematics	3	1	0	40	60	100	4
BTES 401-18	Engineering Science Course	Computer Organization & Architecture	3	0	0	40	60	100	3
BTCS 402-18	Professional Core Courses	Operating Systems	3	0	0	40	60	100	3
BTCS 403-18	Professional Core Courses	Design & Analysis of Algorithms	3	0	0	40	60	100	3
HSMC 122-18	Humanities & Social Sciences including Management Courses	Universal Human Values 2	2	1	0	40	60	100	3
EVS101-18	Mandatory Courses	Environmental Sciences	3	-	-	100	-	100	S/US
BTES 402-18	Engineering Science Course	Computer Organization & Architecture Lab	0	0	2	30	20	50	1
BTCS 404-18	Professional Core Courses	Operating Systems Lab	0	0	4	30	20	50	2
BTCS 405-18	Professional Core Courses	Design & Analysis of Algorithms Lab	0	0	4	30	20	50	2
Total			**15**	**2**	**10**	**290**	**360**	**650**	**24**

Students will take up summer internship of 4-6 weeks at industry or organizations of repute after 4th sem, that will be accredited in 5th semester.

Fifth Semester

Course Code	Type of Course	Course Title	Hours per Week			Marks Distribution		Total Marks	Credits
			L	T	P	Internal	External		
BTCS 501-18	Professional Core Courses	Database Management Systems	3	0	0	40	60	100	3
BTCS 502-18	Professional Core Courses	Formal Language & Automata Theory	3	1	0	40	60	100	3
BTCS 503-18	Professional Core Courses	Software Engineering	3	0	0	40	60	100	3
BTCS 504-18	Professional Core Courses	Computer Networks	3	0	0	40	60	100	3
BTCS XXX-18	Professional Elective	Elective-I	3	0	0	40	60	100	3
BTCS YYY-18	Professional Elective Courses	Elective-II	3	0	0	40	60	100	3
BTCS 505-18	Professional Core Courses	Database Management Systems Lab	0	0	4	30	20	50	2
BTCS 506-18	Professional Core Courses	Software Engineering Lab	0	0	2	30	20	50	1
BTCS 507-18	Professional Core Courses	Computer Networks Lab	0	0	2	30	20	50	1
BTCS XXX-18	Professional Elective	Elective-I Lab	0	0	2	30	20	50	1
BTCS YYY-18	Professional Elective Courses	Elective-II lab	0	0	2	30	20	50	1
		Industrial Training	0	0	0	60	40	100	Satisfactory/Unsatisfactory
Total			**18**	**1**	**12**	**450**	**500**	**950**	**24**

Sixth Semester

Course Code	Type of Course	Course Title	Hours per Week			Marks Distribution		Total Marks	Credits
			L	T	P	Internal	External		
BTCS 601-18UC	Professional Core Courses	Compiler Design	3	0	0	40	60	100	3
BTCS 602-18UC	Professional Core Courses	Artificial Intelligence	3	1	0	40	60	100	3
BTCS ZZZ-18UC	Professional Elective Courses	Elective-III	3	0	0	40	60	100	3
BTCS UUU-18UC	Professional Elective Courses	Elective-IV	3	0	0	40	60	100	3
BTOE ***	Open Elective Courses	Open Elective-I	3	0	0	40	60	100	3
BTCS 603-18UC	Project	Project-1	0	0	6	60	40	100	3
BTCS 604-18UC	Professional Core Courses	Compiler Design Lab	0	0	2	30	20	50	1
BTCS 605-18UC	Professional Core Courses	Artificial Intelligence Lab	0	0	2	30	20	50	1
BTCS ZZZ-18UC	Professional Elective Courses	Elective-III Lab	0	0	2	30	20	50	1
BTCS UUU-18UC	Professional Elective Courses	Elective-IV Lab	0	0	2	30	20	50	1
Total			**15**	**0**	**14**	**380**	**420**	**800**	**22**

Open Elective is to be selected from other departments Scheme.

Seventh Semester

Course Code	Course Type	Course Title	Load Allocations			Marks Distribution		Total Marks	Credits
			L	T	P	Internal	External		
BTCS VVV-18UC	Professional Elective	Elective-V	3	0	0	40	60	100	3
BTCS TTT-18UC	Professional Elective Courses	Elective-VI	3	0	0	40	60	100	3
BTOE ***	Open Elective Courses	Open Elective-II	3	0	0	40	60	100	3
BTOE ***	Open Elective Courses	Open Elective- III	3	0	0	40	60	100	3
BTCS 701-18UC	Professional Core Courses	Machine Learning	3	0	0	40	60	100	3
BTCS 702-18UC	Project	Project-II	0	0	12	120	80	200	6
BTCS 703-18UC	Professional Core Courses	Machine Learning Lab	0	0	2	30	20	50	1
BTCS VVV-18UC	Professional Elective	Elective-V lab	0	0	2	30	20	50	1
BTCS TTT-18UC	Professional Elective Courses	Elective-VI lab	0	0	2	30	20	50	1
Total			**15**	**0**	**18**	**410**	**440**	**850**	**24**

Eighth Semester

Course Code	Course Title	Marks Distribution		Total Marks	Credits
		Internal	External		
BTCS 801-18UC	Semester Training	300	200	500	16

Course Code	Course Type	Course Title	Load Allocations			Marks Distribution		Total Marks	Credits
			L	T	P	Internal	External		
BTCS 801-18UC	Professional Core Courses	Social Network Analysis	2	0	2	40	60	100	3
BTCS 802-18UC	Professional Core Courses	Cyber Attacks	2	0	4	40	60	100	3
BTCS 803-18UC	Professional Core Courses	Deep Learning	3	0	2	40	60	100	4
BTCS 804-18UC	Project	Project-III	0	0	12	120	80	200	6
Total			**7**	**0**	**20**	**240**	**260**	**500**	**16**

LIST OF ELECTIVES

Elective-I

BTCS 508-18	Programming in Java
BTCS 509-18	Web and Open Source Technologies
BTCS 510-18	Programming in Python
BTCS 511-18	Programming in Java lab
BTCS 512-18	Web and Open Source Technologies lab
BTCS 513-18	Programming in Python Lab

Elective-II

BTCS 514-18	Mobile Application Development
BTCS 515-18	Computer Graphics
BTCS 516-18	Internet of Things
BTCS 517-18	Mobile Application Development lab
BTCS 518-18	Computer Graphics Lab
BTCS 519-18	Internet of Things Lab

Elective-III

BTCS 606-18UC	Network Security and Cryptography
BTCS 607-18UC	Data Mining
BTCS 608-18UC	Cloud Computing
BTCS 609-18UC	Network Security and Cryptography Lab
BTCS 610-18UC	Data Mining lab
BTCS 611-18UC	Cloud Computing lab

Elective-IV

BTCS 612-18UC	Information Theory and Coding
BTCS 613-18UC	Data Science
BTCS 614-18UC	Soft Computing
BTCS 615-18UC	Information Theory and Coding lab
BTCS 616-18UC	Data Science Lab
BTCS 617-18UC	Soft Computing lab

Elective-V

BTCS 704-18UC	Quantum Computing
BTCS 705-18UC	Ad-Hoc and Sensor Networks
BTCS 706-18UC	Speech and Natural Language Processing
BTCS 707-18UC	Quantum Computing lab
BTCS 708-18UC	Ad-Hoc and Sensor Networks lab
BTCS 709-18UC	Speech and Natural Language Processing lab

Open electives offered by the department:

Courses of odd semesters:

BTCS301-18	Enterprise Resource Planning
BTCS302-18	Cyber laws and IPR
BTCS501-18	Database Management System
BTCS504-18	Computer Networks
BTCS602-18UC	Artificial Intelligence
BTCS613-18UC	Data Science
BTCS710-18UC	Block Chain Technologies

Courses of even semesters:

BTES401-18	Computer Organisation & Architecture
BTCS402-18	Operating System
BTCS618-18UC	Internet of Things
BTCS619-18UC	Cyber Security

LIST OF COURSES FOR HONOURS DEGREE

In order to have an Honours degree, a student choose 18-20 credits from the following courses in addition.

Course Code	Type of Course	Course Title	Hours per Week			Marks Distribution		Total Marks	Credits
			L	T	P	Internal	External		
BTCS H01-18	Professional Elective Courses	Graph Theory	3	0	0	40	60	100	3
BTCS H02-18	Professional Elective Courses	Computer Vision	3	0	0	40	60	100	3
BTCS 618-18	Professional Elective Courses	Embedded Systems	3	0	0	40	60	100	3
BTCS H03-18	Professional Elective Courses	Software Project Management	3	0	0	40	60	100	3
BTCS H04-18	Professional Elective Courses	Cryptography & Network Security	3	0	0	40	60	100	3
BTCS H05-18	Professional Elective Courses	Internet-of-Things	3	0	0	40	60	100	3
BTCS 805-18	Professional Elective Courses	Data Analytics	3	0	0	40	60	100	3
BTCS 701-18	Professional Elective Courses	Machine Learning	3	0	0	40	60	100	3
BTCS H06-18	Professional Elective Courses	ICT in Agriculture and Rural Development	3	0	0	40	60	100	3
BTCS H07-18	Professional Elective Courses	Computational Technologies for Smart Cities	3	0	0	40	60	100	3
BTCS H08-18	Professional Elective Courses	Computer Forensics	3	0	0	40	60	100	3

MINOR DEGREE IN COMPUTER SCIENCE ENGG.(Credits required 20 from Core+Electives/MOOCS*)

List of Core Courses: Minimum of 2 courses must be opted, other than studied in regular course.

Type of Course	Course Title	Hours per Week			Marks Distribution		Total Marks	Credits
		L	T	P	Internal	External		
PCC	Data structure Theory & Lab	3	0	4	40T+30 P	60T+20 P	150	5
PCC	Object Oriented Programming Theory & Lab	3	0	4	40T+30 P	60T+20 P	150	5
PCC	Computer networks Theory & Lab	3	0	2	40T+30 P	60T+20 P	150	4
PCC	Operating system Theory & Lab	3	0	4	40T+30 P	60T+20 P	150	5
ESC	Computer Organisation and architecture Theory & Lab	3	0	2	40T+30 P	60T+20 P	150	4
PCC	Database Management system Theory & Lab	3	0	4	40T+30 P	60T+20 P	150	5

***For course code refer to scheme given above.**

*List of Courses through MOOCS will be provided every six months through BOS/ MOOCS Coordinator; each course must be of minimum 12 weeks and of 4 credits after submission of successful exam in that course.

List of Electives: 3 courses can be opted, other than studied in regular course.

Type of Course	Course Title	Hours per Week			Marks Distribution		Total Marks	Credits
		L	T	P	Internal	External		
ELECTIVE	Web Technologies Theory & Lab	3	0	2	40T+30 P	60T+20 P	150	4
ELECTIVE	Machine Learning Theory & Lab	3	0	2	40T+30 P	60T+20 P	150	4
ELECTIVE	Cloud computing Theory & Lab	3	0	2	40T+30 P	60T+20 P	150	4
ELECTIVE	Adhoc and Sensor network Theory & Lab	3	0	2	40T+30 P	60T+20 P	150	4
ELECTIVE	Data Analysis Theory & Lab	3	0	2	40T+30 P	60T+20 P	150	4
ELECTIVE	Computer Graphics Theory & Lab	3	0	2	40T+30 P	60T+20 P	150	4
ELECTIVE	Mobile Application Development Theory & Lab	3	0	2	40T+30 P	60T+20 P	150	4
ELECTIVE	Data Mining Theory & Lab	3	0	2	40T+30 P	60T+20 P	150	4
ELECTIVE	Information Theory & Coding Theory & Lab	3	0	2	40T+30 P	60T+20 P	150	4
ELECTIVE	Soft Computing Theory & Lab	3	0	2	40T+30 P	60T+20 P	150	4

***For course code refer to scheme given above.**

Third Semester

Course Code: BTCS301-18	**Course Title:** Data Structure & Algorithms	**3L:0T:0P**	**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 4: 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: BTCS302-18	**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, 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: BTCS303-18	**Course Title:** Data Structure & Algorithms Lab	**0L:0T:4P**	**2Credits**

List of Experiment:

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 following insertion operations in a single linked list:

i. Insertion at 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 following deletion operations in a single linked list:

i. Deletion at beginning
ii. Deletion at 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: BTCS304-18	**Course Title:** Object Oriented Programming Lab	**0L:0T:4P**	**2Credits**

List of Experiment:

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

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, JoseeLajoie: C++ Primer, 4th Edition, Addison Wesley, 2012.
2. E. Balagurusamy, Object Oriented Programming with C++, Tata McGraw Hill.

BTAM304-18	**Mathematics Paper-III (Calculus and Ordinary Differential Equations)**	**3L:0T:0P**	**3 credits**

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.

[CO1, CO2] **(12Hrs)**

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.

[CO3] **(13Hrs.)**

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.

[CO4**] (12 hrs.)**

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.

[CO5] **(12 hrs.)**

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 mostbranches of engineering;
2. Apply multiple integrals to deal with areas and volumes of various structures which are quite significant in real world;
3. Formulate and solveengineering problems related to convergence, infinite series, power series and Taylor series;
4. Create, select and utilize the learnt techniques of first degree ordinary differential equationsto 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.

……

Development of Societies
Course code: HSMC101-18

Credits: 3

COURSE TOPICS:

2.1 Unit I: Social Development (5 hours)

1. Concepts behind the origin of Family, Clan and Society
2. Different Social Systems
3. Relation between Human being and Society
4. Comparative studies on different models of Social Structures and their evolution

2.2 Unit II: Political Development (3 hours)

1. Ideas of Political Systems as learnt from History
2. Different models of Governing system and their comparative study

2.3 Unit III: Economic Development (18 hours)

1. Birth of Capitalism, Socialism, Marxism
2. Concept of development in pre-British, British and post British period- Barter, Jajmani
3. Idea of development in current context.
4. E. F. Schumacher's idea of development, Buddhist economics. Gandhian idea of development. Swaraj and Decentralization.

3. READINGS

3.1 TEXTBOOK:

3.2 *REFERENCE BOOKS:

4. OTHER SESSIONS

4.1 *TUTORIALS:

4.2 *LABORATORY:

4.3 *PROJECT: Possible projects in this course could be

a) Interact with local communities and understand their issues.
b) Study local cottage industry and agricultural practices. Role of engineering and specialized knowledge.
c) Evaluation of technology in the context of its application. Social impact of technology. Environmental impact of technology. Evaluation from a holistic perspective.

PHILOSOPHY
Course code: HSMC102-18

Credits: 3

COURSE TOPICS:

2.1 Unit 1:
The difference between knowledge (Vidya) and Ignorance (Avidya):

a. Upanishads;
b. Six systems orthodox and Heterodox Schools of IndianPhilosophy.
c. Greek Philosophy:

2.2 Unit 2:
Origin of the Universe:

- NasidiyaSukta: "Who reallyknows?"
- Brhadaranyaka Upanishad; Chandogya Upanishad: Non-self, Self, real and unreal.
- Taittiriya Upanishad: SikshaValli.
- Plato's Symposium: Lack as the source of desire andknowledge.
- Socratic's method of knowledge as discovery.
- Language: Word as root of knowledge (Bhartrahari'sVakyapadiyam)
- Fourteen Knowledge basis as a sources of Vidya: Four Vedas; Six auxiliary sciences (Vedangas); Purana, Nyaya, Mimamsa and Dharma Sastras.

2.3 Unit 3:
Knowledge as Power: Francis Bacon. Knowledge as both power and self-realization in Bagavad Gita.

2.4 Unit 4:
Knowledge as oppression: M. Foucault. Discrimination between Rtam and Satyam in Indian Philosophy.

2.5 Unit 5:
Knowledge as invention: Modern definition of creativity; scientific activity in the claim that science invents new things at least through technology.

2.6 Unit 6:
Knowledge about the self, transcendental self; knowledge about society, polity and nature.

2.7 Unit 7:
Knowledge about moral and ethics codes.

2.8 Unit 8:
Tools of acquiring knowledge: Tantrayuktis, a system of inquiry (Caraka, Sushruta, Kautilya, Vyasa)

2. READINGS

1. Copleston, Frederick, History of Philosophy, Vol. 1.Great Britain: Continuum.

2. Hiriyanna, M. Outlines of Indian Philosophy, MotilalBanarsidass Publishers; Fifth Reprint edition (2009)
3. Sathaye, Avinash, Translation of NasadiyaSukta
4. Ralph T. H. Griffith. The Hymns of the Ṛgveda. MotilalBanarsidass: Delhi: 1973.
5. Raju, P. T. Structural Depths of Indian Thought, Albany: State University of New York Press.
6. Plato, Symposium, Hamilton Press.
7. KautilyaArtha Sastra. Penguin Books, New Delhi.
8. Bacon, Nova Orgum
9. Arnold, Edwin. The Song Celestial.
10. Foucault, Knowledge/Power.
11. Wildon, Anthony, System of Structure.
12. Lele, W.K. The Doctrine of Tantrayukti. Varanasi: Chowkamba Series.
13. Dasgupta, S. N. History of Indian Philosophy, MotilalBanasidas, Delhi.
14. Passmore, John, Hundred Years of Philosophy, Penguin.

4. OTHER SESSIONS:

- Mode of Conduct

5. ASSESSMENT (indicative only):

Ask students to do term papers, for example, writing biographical details of founders, sustainers, transmitters, modifiers, rewriters; translating monographs of less known philosophers such as K. C. Bhattacharys, Daya Krishna, Gopinath Bhattacharya; comparative study of philosophical system such as MadhyasthaDarshan.

6. OUTCOME OF THE COURSE:

Students will develop strong natural familiarity with humanities along with right understanding enabling them to eliminate conflict and strife in the individual and society. Students shall be able to relate philosophy to literature, culture, society and lived experience can be considered.

……

Course Code:BTES301-18	**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 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:

- Morris Mano, **Digital Design**, Prentice Hall of India Pvt. Ltd
- Donald P.Leach and Albert Paul Malvino, **Digital Principles and Applications**, 5 ed., Tata

McGraw HillPublishing CompanyLimited, New Delhi, 2003.

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

…….

.

Course Code:BTES302-18	**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

……

Fourth Semester

Course Code: BTES401-18	**Course Title:** Computer Organization & Architecture	**3L:0T:0P**	**3Credits**

Pre-requisites: Digital Electronics

Detailed Contents:

Module 1: Functional blocks of a computer

CPU, memory, input-output subsystems, control unit. Instruction set architecture of a CPU – registers, instruction execution cycle, RTL interpretation of instructions, addressing modes, instruction set. Case study – instruction set of 8085 processor.
Data representation: signed number representation, fixed and floating point representations, character representation. Computer arithmetic – integer addition and subtraction, ripple carry adder, carry look-ahead adder, etc. multiplication – shift-and add, Booth multiplier, carry save multiplier, etc. Division restoring and non-restoring techniques, floating point arithmetic.

[10 hrs] (CO1, CO2)

Module 2: Introduction to x86 architecture.

CPU control unit design: Hardwired and micro-programmed design approaches, Case study – design of a simple hypothetical CPU.
Memory system design: semiconductor memory technologies, memory organization. **Peripheral devices and their characteristics**: Input-output subsystems, I/O device interface, I/O transfers – program controlled, interrupt driven and DMA, 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

Basic concepts of pipelining, throughput and speedup, pipeline hazards.
Parallel Processors: Introduction to parallelprocessors, Concurrent access to memory and cache coherency.

[10 hrs] (CO5)

Module 4: Memory Organization

Memory interleaving, concept of hierarchical memory organization, cache memory, cache size vs. block size, mapping functions, replacement algorithms, write policies.

[10 hrs] (CO3)

Course Outcomes:

The student will be able to:

1. Understand functional block diagram of microprocessor;
2. Apply instruction set for Writingassembly language programs;
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. "ComputerOrganization and Architecture", Moris Mano,
2. "ComputerOrganization 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 CarlHamacher, 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: BTCS402-18	**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: BTCS403-18	**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 Masters' 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, 1st Edition, 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: BTES402-18	**Course Title:** Computer Organization & Architecture Lab	**0L:0T:2P**	**1Credits**

List of Experiment:

Task 1: Computer Anatomy- Memory, Ports, Motherboard and add-on cards.
Task 2: Dismantling and assembling PC.

Task 3: Introduction to 8085 kit.

Task 4:2. Addition of two 8 bit numbers, sum 8 bit.
Task 5: Subtraction of two 8 bit numbers.
Task 6: Find 1's complement of 8-bit number.
Task 7: Find 2's complement of 8-bit number.
Task 8: Shift an 8-bit no. by one bit.
Task 9: Find Largest of two 8 bit numbers.
Task 10: Find Largest among an array of ten numbers (8 bit).
Task 11: Sum of series of 8 bit numbers.

Task 12: Introduction to 8086 kit.

Task 13: Addition and subtraction of two 16 bit numbers, sum 16 bit.
Task 14: Implement of Booth's algorithm for arithmetic operations.
Task 15: Find 1's and 2's complement of 16-bit number.
Task 16: Implement simple programs using I/O based interface.

Lab Outcomes:

The student will be able to:

1. Assemble personal computer;
2. Implement the various assembly language programs for basic arithmetic and logical operations; &
3. Demonstrate the functioning of microprocessor/microcontroller based systems with I/O interface.

Reference Books:

1. Fundamentals of Microprocessors and Microcontrollers by B. Ram, Dhanpat Rai Publications.

……

Course Code: BTCS404-18	**Course Title:** Operating Systems Lab	**0L:0T:4P**	**2Credits**

List of Experiment:

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: BTCS405-18	**Course Title:** Design and Analysis of Algorithms Lab	**0L:0T:4P**	**2Credit**

List of Experiment:

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.

……..

UNIVERSAL HUMAN VALUES 2: UNDERSTANDING HARMONY

Course code: HSMC122-18

Credits: 3

COURSE TOPICS:

The course has 28 lectures and 14 practice sessions in 5 modules:

Module 1: Course Introduction - Need, Basic Guidelines, Content and Process for Value Education

1. Purpose and motivation for the course, recapitulation from Universal Human Values-I
2. Self-Exploration–what is it? - Its content and process; 'Natural Acceptance' and ExperientialValidation- as the process forself-exploration.
3. Continuous Happiness and Prosperity-A look at basic Human Aspirations
4. Right understanding, Relationship and Physical Facility- the basic requirementsfor fulfilment of aspirations of every human being with their correct priority
5. Understanding Happiness and Prosperity correctly- A critical appraisal of the current scenario.
6. Method to fulfil the above human aspirations: understanding and living inharmony at various levels.

Include practice sessions to discuss natural acceptance in human being as the innate acceptance for living with responsibility (living in relationship, harmony and co- existence) rather than as arbitrariness in choice based on liking-disliking.

Module 2: Understanding Harmony in the Human Being - Harmony in Myself!

1. Understanding human being as a co-existence of the sentient 'I' and the material 'Body'
2. Understanding the needs of Self ('I') and 'Body' - happiness and physical facility
3. Understanding the Body as an instrument of 'I' (I being the doer, seer and enjoyer)
4. Understanding the characteristics and activities of 'I' and harmony in 'I'
5. Understanding the harmony of I with the Body: Sanyam and Health; correct appraisal of Physical needs, meaning of Prosperityin detail
6. Programs to ensureSanyam and Health.

Include practice sessions to discuss the role others have played in making material goods available to me. Identifying from one's own life. Differentiate between prosperity and accumulation. Discuss program for ensuring health vs dealing with disease.

Module 3: Understanding Harmony in the Family and Society- Harmony in Human-Human Relationship

1. Understanding values in human-human relationship; meaning of Justice (nine universal values in relationships) and program for its fulfilment to ensure mutual happiness; Trust and Respect as the foundational values ofrelationship.
2. Understanding the meaning of Trust; Difference between intention and competence
3. Understanding the meaning of Respect, Difference between respect and differentiation; the other salient values inrelationship.
4. Understanding the harmony in the society (society being an extension of family): Resolution, Prosperity, fearlessness (trust) and co-existence as comprehensive Human Goals.
5. Visualizing a universal harmonious order in society- Undivided Society,
6. Universal Order- from family to world family.

Include practice sessions to reflect on relationships in family, hostel and institute as extended family, real life examples, teacher-student relationship, goal of education etc. Gratitude as a universal value in relationships. Discuss with scenarios. Elicit examples from students' lives.

Module 4: Understanding Harmony in the Nature and Existence - Whole existence as Coexistence

1. Understanding the harmony in the Nature
2. Interconnectedness and mutual fulfilment among the four orders of nature - recyclability and self-regulation in nature
3. Understanding Existence as Co-existence of mutually interacting units in all- pervasive space
4. Holistic perception of harmony at all levels of existence.

Include practice sessions to discuss human being as cause of imbalance in nature (film "Home" can be used), pollution, depletion of resources and role of technology etc.

Module 5: Implications of the above Holistic Understanding of Harmony on Professional Ethics

1. Natural acceptance of human values
2. Definitiveness of Ethical Human Conduct
3. Basis for Humanistic Education, Humanistic Constitution and Humanistic Universal Order
4. Competence in professional ethics: a. Ability to utilize the professional competence for augmenting universal human order b. Ability to identify the scope and characteristics of peoplefriendly and eco -friendly production systems, c. Ability to identify and develop appropriatetechnologies and management patterns for above production systems.
5. Case studies of typical holistic technologies, management models and production systems.
6. Strategy for transition from the present state to Universal Human Order: a. At the level of individual: as socially and ecologically responsible engineers, technologists and managers b. At the level of society: as mutually enriching institutions and organizations.
7. Sum up.

Include practice Exercises and Case Studies will be taken up in Practice (tutorial) Sessions eg. to discuss the conduct as an engineer or scientist etc.

3. READINGS:

3.1 Text Book

1. Human Values and Professional Ethics by R R Gaur, R Sangal, G P Bagaria, Excel Books, New Delhi, 2010.

3.2 Reference Books

1. Jeevan Vidya: EkParichaya, A. Nagaraj, Jeevan VidyaPrakashan, Amarkantak, 1999.
2. Human Values, A.N. Tripathi, New Age Intl. Publishers,New Delhi, 2004.
3. The Story of Stuff (Book).
4. The Story of My Experiments with Truth - by Mohandas Karamchand Gandhi
5. Small is Beautiful - E. F Schumacher.
6. Slow is Beautiful - Cecile Andrews
7. Economy of Permanence - J CKumarappa
8. Bharat Mein Angreji Raj -PanditSunderlal
9. Rediscovering India - by Dharampal
10. Hind Swaraj or Indian Home Rule - by Mohandas K. Gandhi
11. India Wins Freedom - Maulana Abdul Kalam Azad
12. Vivekananda - Romain Rolland (English)
13. Gandhi - Romain Rolland (English)

OUTCOME OF THE COURSE:

By the end of the course, students are expected to become more aware of themselves, and their surroundings (family, society, nature); they would become more responsible in life, and in handling problems with sustainable solutions, while keeping human relationships and human nature in mind. They would have better critical ability. They would also become sensitive to their commitment towards what they have understood (human values, human relationship and human society). It is hoped that they would be able to apply what they have learnt to their own self in different day-to-day settings in real life, at least a beginning would be made in this direction.

This is only an introductory foundational input. It would be desirable to follow it up by

a) Faculty -student or mentor-mentee programs throughout their time with the institution.
b) Higher level courses on human values in every aspect of living. E.g. as a professional.

……

Course Code: EVS101-18	Course Title: Environmental Studies	3L:0T:0P	0Credits

Detailed Contents

Module 1 : Natural Resources :Renewable and non-renewable resources

Natural resources and associated problems.

a) Forest resources : Use and over-exploitation, deforestation, case studies. Timber extraction, mining, dams and their effects on forest and tribal people.
b) Water resources : Use and over-utilization of surface and ground water, floods, drought, conflicts over water, dams-benefits and problems.
c) Mineral resources : Use and exploitation, environmental effects of extracting and using mineral resources, case studies.
d) Food resources : World food problems, changes caused by agriculture and overgrazing, effects of modern agriculture, fertilizer-pesticide problems, water logging, salinity, case studies.
e) Energy resources : Growing energy needs, renewable and non renewable energy sources, use of alternate energy sources. Case studies.
f) Land resources : Land as a resource, land degradation, man induced landslides, soil erosion and desertification.

- Role of an individual in conservation of natural resources.
- Equitable use of resoureces for sustainable lifestyles.

Module 2 : Ecosystems

Concept of an ecosystem.

Structure and function of an ecosystem.

Food chains, food webs and ecological pyramids. Introduction, types, characteristic features, structure and function of following ecosystems:

a. Forest ecosystem
b. Aquatic ecosystems (ponds, streams, lakes, rivers, oceans, estuaries)

Module 3 : Biodiversity and its conservation

- Introduction – Definition : genetic, species and ecosystem diversity.
- Biodiversity at global, National and local levels.
- Inida as a mega-diversity nation
- Hot-sports of biodiversity.
- Threats to biodiversity : habitat loss, poaching of wildlife, man-wildlife conflicts.
- Endangered and endemic species of India

Module 4 : Social Issues and the Environment

- From Unsustainable to Sustainable development
- Resettlement and rahabilitation of people; its problems and concerns.
- Environmental ethics : Issues and possible solutions.
- Climate change, global warming, acid rain, ozone layer depletion, Nuclear accidents and holocaust. Case Studies.
- Public awareness.

*ACTIVITIES

Nature club (bird watching, recognizing plants at institute/at home, recognizing local animals, appreciating biodiversity

Impart knowledge and inculcate the habit of taking interest and understanding biodiversity in and around the college campus. The students should be encouraged to take interest in bird watching, recognizing local plants, herbs and local animals. The students should be encouraged to appreciate the difference in the local

biodiversity in their hometown, in the place of their study and other places they visit for vacation/breaks etc.
Following activities must be included.
Identify a tree fruit flower peculiar to a place or having origin from the place.
Making high resolution big photographs of small creatures (bees, spiders, ants. mosquitos etc.) especially part of body so that people can recognize (games on recognizing animals/plants).
Videography/ photography/ information collections on specialties/unique features of different types of common creatures.
Search and explore patents and rights related to animals, trees etc. Studying miracles of mechanisms of different body systems.

1*(A) Awareness Activities:*

a) Small group meetings about water management, promotion of recycle use, generation of less waste, avoiding electricity waste
b) Slogan making event
c) Poster making event
d) Cycle rally
e) Lectures from experts
f) Plantation
g) Gifting a tree to see its full growth
h) Cleanliness drive
i) Drive for segregation of waste
i) To live with some eminent environmentalist for a week or so to understand his work
i. To work in kitchen garden for mess
j) To know about the different varieties of plants
k) Shutting down the fans and ACs of the campus for an hour or so
l) Visit to a local area to document environmental assets river/forest/grassland/hill/mountain/lake/Estuary/Wetlands
m) Visit to a local polluted site-Urban/Rural/Industrial/Agricultural n) Visit to a Wildlife sanctuary, National Park or Biosphere Reserve

Suggested Readings

1. Agarwal, K.C. 2001 Environmental Biology, Nidi Publ. Ltd. Bikaner.
2. BharuchaErach, The Biodiversity of India, Mapin Publishing Pvt. Ltd., Ahmedabad – 380 013, India, Email:mapin@icenet.net (R)
3. Brunner R.C., 1989, Hazardous Waste Incineration, McGraw Hill Inc. 480p
4. Clark R.S., Marine Pollution, Clanderson Press Oxford (TB)
5. Cunningham, W.P. Cooper, T.H. Gorhani, E & Hepworth, M.T. 2001, Environmental Encyclopedia, Jaico Publ. House, Mumabai, 1196p
6. Hawkins R.E., Encyclopedia of Indian Natural History, Bombay Natural History Society, Bombay (R)
7. Heywood, V.H &Waston, R.T. 1995. Global Biodiversity Assessment. Cambridge Univ. Press 1140p.
8. Mhaskar A.K., Matter Hazardous, Techno-Science Publication (TB)
9. Miller T.G. Jr. Environmental Science, Wadsworth Publishing Co. (TB)
10. Odum, E.P. 1971. Fundamentals of Ecology. W.B. Saunders Co. USA, 574p
11. Townsend C., Harper J, and Michael Begon, Essentials of Ecology, Blackwell Science (TB)
12. Trivedi R.K., Handbook of Environmental Laws, Rules Guidelines, Compliances and Stadards, Vol I and II, Enviro Media (R)
13. Trivedi R. K. and P.K. Goel, Introduction to air pollution, Techno-Science Publication (TB)
14. Wanger K.D., 1998 Environmental Management. W.B. Saunders Co. Philadelphia, USA 499p

…….

Course Code: HSMC101- 18	**Course Title: Development of Societies**	**3L:0T:0P**	**3Credits**

Detailed Contents:

Unit I: Social Development (5 hours)

1. Concepts behind the origin of Family, Clan and Society
2. Different Social Systems
3. Relation between Human being and Society
4. Comparative studies on different models of Social Structures and their evolution

Unit II: Political Development (3 hours)

1. Ideas of Political Systems as learnt from History
2. Different models of Governing system and their comparative study

Unit III: Economic Development (18 hours)

1. Birth of Capitalism, Socialism, Marxism
2. Concept of development in pre-British, British and post British period- Barter, Jajmani
3. Idea of development in current context.
4. E. F. Schumacher's idea of development, Buddhist economics. Gandhian idea of development. Swaraj and Decentralization.

PROJECT: Possible projects in this course could be

a) Interact with local communities and understand their issues.
b) Study local cottage industry and agricultural practices. Role of engineering and specialized knowledge.
c) Evaluation of technology in the context of its application. Social impact of technology. Environmental impact of technology. Evaluation from a holistic perspective.

…….

Course Code: HSMC102-18	Course Title: PHILOSOPHY	3L:0T:0P	3Credits

Detailed Contents:

Unit 1:

The difference between knowledge (Vidya) and Ignorance (Avidya):

a. Upanishads;
b. Six systems orthodox and Heterodox Schools of Indian Philosophy.
c. Greek Philosophy:

Unit 2:

Origin of the Universe:

- NasidiyaSukta: "Who really knows?"
- Brhadaranyaka Upanishad; Chandogya Upanishad: Non-self, Self, real and unreal.
- Taittiriya Upanishad: SikshaValli.
- Plato's Symposium: Lack as the source of desire and knowledge.
- Socratic's method of knowledge as discovery.
- Language: Word as root of knowledge (Bhartrahari'sVakyapadiyam)
- Fourteen Knowledge basis as a sources of Vidya: Four Vedas; Six auxiliary sciences (Vedangas); Purana, Nyaya, Mimamsa and Dharma Sastras.

Unit 3:

Knowledge as Power: Francis Bacon. Knowledge as both power and self-realization in Bagavad Gita.

Unit 4:

Knowledge as oppression: M. Foucault. Discrimination between Rtam and Satyam in Indian Philosophy.

Unit 5:

Knowledge as invention: Modern definition of creativity; scientific activity in the claim that science invents new things at least through technology.

Unit 6:

Knowledge about the self, transcendental self; knowledge about society, polity and nature.

Unit 7:

Knowledge about moral and ethics codes.

Unit 8:

Tools of acquiring knowledge: Tantrayuktis, a system of inquiry (Caraka, Sushruta, Kautilya, Vyasa)

READINGS

a. Copleston, Frederick, History of Philosophy, Vol. 1.Great Britain: Continuum.
2. Hiriyanna, M. Outlines of Indian Philosophy, MotilalBanarsidass Publishers; Fifth Reprint edition (2009)
3. Sathaye, Avinash, Translation of NasadiyaSukta
4. Ralph T. H. Griffith. The Hymns of the Ṛgveda. MotilalBanarsidass: Delhi: 1973.
5. Raju, P. T. Structural Depths of Indian Thought, Albany: State University of New York Press.
6. Plato, Symposium, Hamilton Press.
7. KautilyaArtha Sastra. Penguin Books, New Delhi.
8. Bacon, Nova Orgum
9. Arnold, Edwin. The Song Celestial.

10. Foucault, Knowledge/Power.
11. Wildon, Anthony, System of Structure.
12. Lele, W.K. The Doctrine of Tantrayukti. Varanasi: Chowkamba Series.
13. Dasgupta, S. N. History of Indian Philosophy, MotilalBanasidas, Delhi.
14. Passmore, John, Hundred Years of Philosophy, Penguin.

ASSESSMENT (indicative only):
Ask students to do term papers, for example, writing biographical details of founders, sustainers, transmitters, modifiers, rewriters; translating monographs of less known philosophers such as K. C. Bhattacharys, Daya Krishna, Gopinath Bhattacharya; comparative study of philosophical system such as MadhyasthaDarshan.

OUTCOME OF THE COURSE:
Students will develop strong natural familiarity with humanities along with right understanding enabling them to eliminate conflict and strife in the individual and society. Students shall be able to relate philosophy to literature, culture, society and livedexperience can be considered.

…….
.

Course Code:BTCS401-18	Course Title: Discrete Mathematics	3L:1T:0P	4 Credits

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 Operation, Rings, Integral Domain and Fields. Boolean Algebra and Boolean Ring, Identitiesof 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.

…….

Fifth Semester

Course Code: BTCS501-18	Course Title: Database Management Systems	3L:0T:0P	3 Credits

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](CO 1, 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](CO 2,4)

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

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](CO 5, 6)**

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 4, 6)**

Course Outcomes:

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

CO 1: For a given query write relational algebra expressions for that query and optimize the developed expressions

CO 2: For a given specification of the requirement design the databases using ER method and normalization.

CO 3: For a given specification construct the SQL queries for Open source and Commercial DBMS - MYSQL, ORACLE, and DB2.

CO 4: For a given query optimize its execution using Query optimization algorithms

CO 5: For a given transaction-processing system, determine the transaction atomicity, consistency, isolation, and durability.

CO 6: Implement the isolation property, including locking, time stamping based on concurrency control and Serializability of scheduling.

Suggested Books:

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

Suggested reference books

1 **"Principles of Database and Knowledge – Base Systems"**, Vol 1 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 : BTCS 502-18	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. 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. 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. Context-sensitive languages: Context-sensitive grammars (CSG) and languages, linear bounded automata and equivalence with CSG. 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. Undecidability: Church-Turing thesis, universal Turing machine, the universal and diagonalization languages, reduction between languages and Rice s theorem, undecidable problems about languages.

[4hrs](CO 1)

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 off finite automata.

[8hrs](CO 2, 3)

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](CO 4, 5)**

Module 4: Context-sensitive languages Context-sensitive grammars (CSG) and languages, linear bounded automata and equivalence with CSG. **[6hrs](CO 5)**

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 5.6)**

Module6: Undecidability Church-Turing thesis, universal Turing machine, the universal and diagonalization languages, reduction between languages and Rices theorem, undecidable problems about languages. **[8hrs](CO 7)**

Course Outcomes:

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

CO 1: Write a formal notation for strings, languages and machines.

CO 2: Design finite automata to accept a set of strings of a language.

CO 3: For a given language determine whether the given language is regular or not.

CO 4: Design context free grammars to generate strings of context free language.

CO 5: Determine equivalence of languages accepted by Push Down Automata and languages generated by context free grammars

CO 6: Write the hierarchy of formal languages, grammars and machines.

CO 7: Distinguish between computability and non-computability and Decidability and undecidability.

Suggested books

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

Suggested 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: BTCS 503-18	Course Title : Software Engineering	3L:0T:0P	3 Credits

Detailed Contents:

UNIT 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.

[8hrs] (CO 1)

UNIT 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.

[6hrs] (CO 2)

UNIT 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.

[8 hrs] (CO 3)

UNIT 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

[8 hrs] (CO 4)

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

[6 hrs] (CO 5)

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: Analyze 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

Suggested Readings/ Books:

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

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

Course Code: BTCS 504 -18	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.

[8 hrs] (CO 1)

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, Goback–NARQ, Selective Repeat ARQ, Sliding Window, Piggy backing, Random Access, Multiple access protocols- Pure ALOHA, Slotted ALOHA, CSMA/CDCDMA/CA.

[10 hrs] (CO 2)

Module 3: Network Layer

Switching, Logical addressing – IPV4, IPV6; Address mapping – ARP, RARP, BOOTP and DHCP – Delivery, Forwarding and Unicast Routing protocols.

[8 hrs] (CO 3)

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] (CO 3)

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] (CO 4)

Course Outcomes:

The student will be able to:

CO 1: Explain the functions of the different layer of the OSI Protocol

CO 2: Describe the function of each block of wide-area networks (WANs), local area networks (LANs) and Wireless LANs (WLANs);

CO 3: Develop the network programming for a given problem related TCP/IP protocol

CO 4: Configure DNS DDNS, TELNET, EMAIL, File Transfer Protocol (FTP), WWW, HTTP, SNMP, Bluetooth, Firewalls using open source available software and tools.

Suggested 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. **Internet working with TCP/IP**, Volume1, 6th Edition Douglas Comer, Prentice Hall of India.
3. **TCP/IP Illustrated**, Volume1, W. Richard Stevens, Addison-Wesley, United States of America.

Elective-I

Course Code: BTCS 508-18	Course Title: Programming in JAVA	3L:0T:0P	3 Credits

Detailed Contents:

Unit 1:

Overview: Object oriented programming principles, Java essentials, java virtual machine, program structure in java

Java class libraries, Data types, Variables and Arrays, Data types and casting, automatic type promotion in expressions, arrays.

Operators and Control Statements: Arithmetic operators, bit wise operators, relational operators, Boolean logical operators, the? Operator, operator precedence

Java's selection statements, iteration statements, jump statements.

CO 1

UNIT 2:

Introduction to Classes: Class fundamentals, declaring class, creating objects

Introducing methods: method declaration, overloading, using objects as parameters, recursion

Constructors, this keyword, garbage collection, the finalization

CO 1

UNIT 3:

Inheritance: Inheritance basics, using super and final, method overriding, dynamic method dispatch, Abstract Class

Interface: variables and extending Interfaces

Package: Creating and importing packages, Package access protection,

Exception Handling: Exception handling fundamentals, Exception types, Uncaught Exceptions Using try and catch, multiple catch clauses, nested try statements, throw, Java’s built-in exceptions.

CO 1,2

UNIT 4:

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

CO 3

UNIT5:

I/O: I/O Basics, Reading Console Input, Writing Console Output, Reading and Writing Files

Applets: Applet Fundamentals, Applet Architecture, The HTML Applet tag, Passing parameters to Applets.

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

CO 4

Course Outcomes:

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

CO1: Understand the features of Java such as opeartors, classes, objects, inheritance, packages and exception handling

CO2: Learn latest features of Java like garbage collection, Console class, Network interface, APIs

CO3: Acquire competence in Java through the use of multithreading, applets

CO4: Get exposure to advance concepts like socket and database connectivity.

Suggested Readings/Books

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

Course Code: BTCS 509-18	Course Title: Web and Open Source Technologies	3L:0T:0P	3 Credits

Detailed Syllabus:

Introduction to WWW: Protocols and programs, secure connections, application and development tools, the web browser, Server, choices, setting up UNIX and Linux web servers, Logging users, dynamic IP

Web Design: Web site design principles, planning the site and navigation

Introduction to HTML: The development process, Html tags and simple HTML forms, web site structure

Introduction to XHTML: XML, Move to XHTML, Meta tags, Character entities, frames and frame sets, inside browser.

Style sheets : Need for CSS, introduction to CSS, basic syntax and structure, using CSS, background images, colors and properties, manipulating texts, using fonts, borders and boxes, margins, padding lists, positioning using CSS, CSS2

JavaScript: Client side scripting, Javascript, How to develop Javascript, simple Javascript, variables, functions, conditions, loops and repetition.

Advance script, Javascript and objects, Javascript own objects, the DOM and web browser environments, forms and validations

DHTML: Combining HTML, CSS and Javascript, events and buttons, controlling your browser

CO 1

Ajax: Introduction, HTTP request, XHttpResponse, AJAX Server Script, AJAX Database, Advantages & disadvantages, Purpose of it, Ajax based web application, alternatives of Ajax

XML: Introduction to XML, uses of XML, simple XML and XML key components, DTD and Schemas, Well formed, using XML with application.XML, XSL and XSLT. Introduction to XSL, XML transformed simple example, XSL elements, transforming with XSLT **CO 2**

PHP: Starting to script on server side, syntax, statements, operators, Arrays, function and forms sessions, E-mail, PHP and AJAX, advance PHP

MySQL Databases : Basic command with PHP examples, Connection to server, creating database, selecting a database, listing database, listing table names creating a table, inserting data, altering tables, queries, deleting database, deleting data and tables, PHPmyadmin and database bugs.

JavaScript Library & Web-Framework:

Jquery: Introduction, Why jQuery, jQuery methods for DOM manipulation, jQuery methods for CSS manipulation, jQuery AJAX Methods (Asynchronous JavaScript and XML)

AngularJS: Fundamental structural concepts of AngularJS, AngularJS Directives, AngularJS Expressions, Use of custom attributes in HTML, introduction to modules and controllers, form validation using validation rules, Server Communication & Data Binding techniques. **CO 3**

Course Outcomes:

At the end of the course the student should be able to:
CO 1: Students are able to develop a dynamic webpage by the use of java script and DHTML.
CO 2: Students will be able to write a well formed / valid XML document.

CO 3: Students will be able to write a server side java application called JSP to catch form data sent from client and store it on database

Suggested Readings/Books:

1. Deitel, Deitel, Nieto, and Sandhu: **XML How to Program**, Pearson Education.
2. Herbert Schildt: **Java 2: The Complete Reference**, Fifth Edition, TMH.
3. Ivan Bayross: **Web Enabled Commercial Application.**
4. Schafer: **Development**, BPB.
5. **HTML, CSS, Java Script,Perl, Python and PHP**, Wiley India Textbooks.
6. R. Peterson, 2007, **Linux: The Complete Reference**, Sixth Edition, TMH.

Course Code: BTCS 510-18	Course Title: Programming in Python	3L:0T:0P	3 Credits

Detailed Syllabus:

UNIT - I 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

CO 1,2

UNIT - II 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

CO 2,3

UNIT - III 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

CO 3,4

UNIT - IV 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

CO 4,5

UNIT – V Database Programming: Introduction, Python Database Application Programmer's Interface (DB-API), Object Relational Managers (ORMs), Related Modules

CO 5

Course Outcomes:

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

CO 1: Examine Python syntax and semantics and be fluent in the use of Python flow control and functions.

CO 2: Demonstrate proficiency in handling Strings and File Systems.

CO 3: Create, run and manipulate Python Programs using core data structures like Lists, Dictionaries and use Regular Expressions.

CO 4: Interpret the concepts of Object-Oriented Programming as used in Python.

CO 5: Implement exemplary applications related to Network Programming, Web Services and Databases in Python.

Suggested Readings/Books

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

Elective-II

Course Code: BTCS 514-18	Course Title: Mobile Application Development	3L:0T:0P	3 Credits

Detailed Syllabus:

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

CO 1

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

CO 1, 2

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

CO 2,3

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

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

CO 4,5

Unit-VI

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

CO 4

Unit-VII

Deployment of apps: Versioning, signing and packaging mobile apps, distributing apps on market place.

CO 5

Course Outcomes:

At the end of the course the student should be able to:
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. T1.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,WroxPublisher,"**Professional Mobile Application Development**", Ist Edition, 2012.
5. Reto Meier, "**Professional Android 4 Application Development**", Wrox, 2012.

Course Code: BTCS 515-18	Course Title: Computer Graphics	3L:0T:0P	3 Credits

Detailed Syllabus:

UNIT–I

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.

CO 1

Unit- II

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.

CO 2

Unit- III

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

CO 2

UNIT–IV

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.

CO 3, 4

Unit- V

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

CO 4, 5

Unit- VI

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.

CO 5

Unit –VII

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.

CO 5, 6

UNIT–VIII

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

CO 6

Course Outcomes:

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

CO 1. To list the concepts used in computer graphics.

CO 2. To implement various algorithms to scan, convert the basic geometrical primitives, transformations, Area filling, clipping.

CO 3. To describe the importance of viewing and projections.

CO 4. To define the fundamentals of animation, virtual reality and its related technologies.

CO 5. To understand a typical graphics pipeline

CO 6. To design an application with the principles of virtual reality

References:

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

Course Code: BTCS 516-18	Course Title: Internet of Things	3L:0T:0P	3Credits

Detailed Syllabus:

1. Introduction to IoT (8 Hours)

Architectural Overview, Design principles and needed capabilities, IoT Applications, Sensing, Actuation, Basics of Networking, M2M and IoT Technology Fundamentals- Devices and gateways, Data management, Business processes in IoT, Everything as a Service(XaaS), Role of Cloud in IoT, Security aspects in IoT.

CO 1

2. Elements of IoT (9 Hours)

Hardware Components- Computing (Arduino, Raspberry Pi), Communication, Sensing, Actuation, I/O interfaces.

Software Components- Programming API's (using Python/Node.js/Arduino) for Communication CO2

Protocols-MQTT, ZigBee, Bluetooth, CoAP, UDP, TCP.

CO 2

3. IoT Application Development (18 Hours)

Solution framework for IoT applications- Implementation of Device integration, Data acquisition and integration, Device data storage- Unstructured data storage on cloud/local server, Authentication, authorization of devices.

CO 3

4. IoT Case Studies (10 Hours)

IoT case studies and mini projects based on Industrial automation, Transportation, Agriculture, Healthcare, Home Automation

CO 4

Course Outcomes:

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

CO 1. To understand internet of Things and its hardware and software components

CO 2. To develop an Interface I/O devices, sensors & communication modules

CO 3. To remotely monitor data and control devices

CO 4. To develop real life IoT based projects

LIST OF SUGGESTED BOOKS

1. Vijay Madisetti, Arshdeep Bahga, **Ïnternet of Things**, "A Hands on Approach", University Press.
2. Dr. SRN Reddy, Rachit Thukral and Manasi Mishra, "**Introduction to Internet of Things: A practical Approach**", ETI Labs.
3. Pethuru Raj and Anupama C. Raman, "**The Internet of Things: Enabling Technologies, Platforms, and Use Cases**", CRC Press.
4. Jeeva Jose, "**Internet of Things**", Khanna Publishing House, Delhi.
5. Adrian McEwen, "**Designing the Internet of Things**", Wiley.

6. Raj Kamal, "**Internet of Things: Architecture and Design**", McGraw Hill.
7. Cuno Pfister, "**Getting Started with the Internet of Things**", O Reilly Media.

Course Code: BTCS 505-18	Course Title: Database management System lab	0L:0T:4P	2 Credits

Detailed List of Tasks:

1. Introduction to SQL and installation of SQL Server / Oracle.
2. Data Types, Creating Tables, Retrieval of Rows using Select Statement, Conditional Retrieval of Rows, Alter and Drop Statements.
3. Working with Null Values, Matching a Pattern from a Table, Ordering the Result of a Query, Aggregate Functions, Grouping the Result of a Query, Update and Delete Statemets.
4. Set Operators, Nested Queries, Joins, Sequences.
5. Views, Indexes, Database Security and Privileges: Grant and Revoke Commands, Commit and Rollback Commands.
6. PL/SQL Architecture, Assignments and Expressions, Writing PL/SQL Code, Referencing Non-SQL parameters.
7. Stored Procedures and Exception Handling.
8. Triggers and Cursor Management in PL/SQL.

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

Course Outcomes:

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: BTCS506-18	Course Title: Software Engineering Lab	0L:0T:2P	1Credits

Detailed List of Tasks:

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

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

Course Code: BTCS507-18	Course Title: Computer Networks Lab	0L:0T:2P	1Credits

Detailed List of Tasks:

Task1: To study the different types of Network cables and network topologies

Task2: Practically implement and test the cross-wired cable and straight through cable using clamping tool and network lab cable tester.

Task3: Study and familiarization with various network devices.

Task4: Familiarization with Packet Tracer Simulation tool/any other related tool. Task5: Study and

Implementation of IP Addressing Schemes

Task6: Creation of Simple Networking topologies using hubs and switches

Task7: Simulation of web traffic in Packet Tracer

Task8: Study and implementation of various router configuration commands

Task9: Creation of Networks using routers.

Task10: Configuring networks using the concept of subnetting

Task11: Practical implementation of basic network command and Network configuration commands like ping, ipconfig, netstat, tracert etc. for trouble shooting network related problems.

Task12: Configuration of networks using static and default routes.

Course Outcomes:

The students will be able to

1. Know about the various networking devices, tools and also understand the implementation of network topologies.
2. Create various networking cables and know how to test these cables.
3. Create and configure networks in packet tracer tool using various network devices and topologies.
4. Understand IP addressing and configure networks using the subnettin.
5. Configure routers using various router configuration commands.
6. Trouble shoot the networks by using various networking commands. Graphics Software's.

Elective-I Lab

Course Code: BTCS511-18	Course Title: Programming in Java Lab	0L:0T:2P	1Credits

To accomplish CO1;

1. WAP in Java to show implementation of classes.
2. WAP in Java to show implementation of inheritance.
3. WAP in Java to show Implementation of packages and interfaces.

To accomplish CO2;

4. WAP in Java to show Implementation of threads.
5. WAP in Java Using exception handling mechanisms.
6. WAP in Java to show Implementation of Applets.

To accomplish CO3;

7. WAP in Java to show Implementation of mouse events, and keyboard events.
8. WAP in Java to show Implementing basic file reading and writing methods.
9. Using basic networking features, WAP in Java

To accomplish CO4;

10. WAP in Java to show Connecting to Database using JDBC.

Project work: A desktop based application project should be designed and implemented in java.

Course Outcomes:

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

CO1. Implement the features of Java such as opeartors, classes, objects, inheritance, packages and exception handling

CO2. Design problems using latest features of Java like garbage collection, Console class, Network interface, APIs

CO3. Develop competence in Java through the use of multithreading, Applets etc

CO4. Apply advance concepts like socket and database connectivity, and develop project based on industry orientation.

Suggested Readings/Books

1. Herbert Schildt, The Complete Reference Java2, McGraw-Hill.

2. Deitel and Deitel, Java: How to Program, 6th Edition, Pearson Education.

3. James Edward Keogh, Jim Keogh, J2EE: The complete Reference, Mc Graw Hill

Course Code: BTCS 512-18	**Course Title: Web and Open Source Technologies Laboratory**	**0L:0T:2P**	**1Credits**

Detailed List of Tasks:

1. Write an HTML page including javascript that takes a given set of integer numbers and shows them after sorting in descending order.
2. Write an HTML page that has one input, which can take multi-line text and a submit button. Once the user clicks the submit button, it should show the number of characters, words and lines in the text entered using an alert message. Words are separated with white space and lines are separated with new line character.
3. Write an HTML page that contains a selection box with a list of 5 countries. When the user selects a country, its capital should be printed next to the list. Add CSS to customize the properties of the font of the capital (color, bold and font size).
4. Create an XML document that contains 10 users information.
5. Using jQuery find all children in a specified class of a division
6. Find all elements of a form that are disabled
7. Create an input form and validate using jQuery. Highlight inputs elements if errors occur
8. Build a Single Page Application (SPA) using AngularJS.

Course Code: BTCS 513-18	Course Title: Programming in Python Lab	0L:0T:2P	1Credits

Detailed List of Tasks:

1. Write a program to demonstrate different number data types in Python.
2. Write a program to perform different Arithmetic Operations on numbers in Python. 3. Write a program to create, concatenate and print a string and accessing sub-string from a given string.
3. Write a python script to print the current date in the following format "Sun May 29 02:26:23 IST 2017"
4. Write a program to create, append, and remove lists in python.
5. Write a program to demonstrate working with tuples in python.
6. Write a program to demonstrate working with dictionaries in python.
7. Write a python program to find largest of three numbers.
8. Write a Python program to convert temperatures to and from Celsius, Fahrenheit. [Formula: c/5 = f-32/9]
9. Write a Python program to construct the following pattern, using a nested for loop

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

10. Write a Python script that prints prime numbers less than 20.
11. Write a python program to find factorial of a number using Recursion.
12. 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).
13. Write a python program to define a module to find Fibonacci Numbers and import the module to another program.
14. Write a python program to define a module and import a specific function in that module to another program.
15. 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.
16. Write a program that inputs a text file. The program should print all of the unique words in the file in alphabetical order.
17. Write a Python class to convert an integer to a roman numeral.
18. Write a Python class to implement pow(x, n)
19. Write a Python class to reverse a string word by word.

Elective-II Lab

Course Code: BTCS 517-18	Course Title: Mobile Application Development Lab	0L:0T:2P	1Credits

Detailed List of Tasks:

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

Course Code: BTCS518-18	**Course Title: Computer Graphics Lab**	**0L:0T:2P**	**1Credits**

Detailed List of Tasks:

1. WAP to draw different geometric structures using different functions.
2. Implement DDA line generating algorithm.
3. Implement Bresenham's line generating algorithm.
4. Implement Mid-point circle line generating algorithm.
5. Implementation of Bresenham's circle drawing algorithm.
6. Implementation of mid-point circle generating Algorithm.
7. Implementation of ellipse generating Algorithm.
8. WAP of color filling the polygon using Boundary fill and Flood fill algorithm.
9. To translate an object with translation parameters in X and Y directions.
10. To scale an object with scaling factors along X and Y directions.
11. Program of line clipping using Cohen-Sutherland algorithm.
12. To perform composite transformations of an object.
13. To perform the reflection of an object about major

Course Code: BTCS 519-18	**Course Title: Internet of Things Lab**	**0L:0T:2P**	**1Credits**

Detailed List of Tasks:

1. Familiarization with Arduino/Raspberry Pi and perform necessary software installation.
2. To interface LED/Buzzer with Arduino/Raspberry Pi and write a program to turn ON LED for 1 sec after every 2 seconds.
3. To interface Push button/Digital sensor (IR/LDR) with Arduino/Raspberry Pi and write a program to turn ON LED when push button is pressed or at sensor detection.
4. To interface DHT11 sensor with Arduino/Raspberry Pi and write a program to print temperature and humidity readings.
5. To interface motor using relay with Arduino/Raspberry Pi and write a program to turn ON motor when push button is pressed.
6. To interface OLED with Arduino/Raspberry Pi and write a program to print temperature and humidity readings on it.
7. To interface Bluetooth with Arduino/Raspberry Pi and write a program to send sensor data to smartphone using Bluetooth.
8. To interface Bluetooth with Arduino/Raspberry Pi and write a program to turn LED ON/OFF when '1'/'0' is received from smartphone using Bluetooth.
9. Write a program on Arduino/Raspberry Pi to upload temperature and humidity data to things peak cloud.
10. Write a program on Arduino/Raspberry Pi to retrieve temperature and humidity data from thing speak cloud.
11. To install MySQL database on Raspberry Pi and perform basic SQL queries.
12. Write a program on Arduino/Raspberry Pi to publish temperature data to MQTT broker.

Sixth Semester

Course Code: BTCS601-18UC	Course Title : Compiler Design	3L:0T:0P	3 Credits

Detailed Contents:

UNIT 1: Introduction: Compiler structure: analysis-synthesis model of compilation, phases of a compiler**.**

Lexical Analysis: Interface with input buffer, parser and symbol table. Token, lexeme and patterns. Difficulties, error reporting and implementation

[8hrs] (CO 1,2)

UNIT 2: Context-free language: Context-free language and grammar, push-down automata, LL(1) grammar, ambiguity, associativity, precedence.
Syntax analysis: Top down parsing, recursive descent parsing, transformation on the grammars, predictive parsing, bottom up parsing, LR parsers: SLR, LALR and LR. Error recovery of parsers.

12hrs] (CO1, 2)

UNIT 3: Semantic Analysis: Syntax directed definitions, inherited and synthesized attributes, dependency graph, evaluation order and evaluation of attributes, L and S attribute.

[6hrs] (CO 1,2)

UNIT 4: Symbol Table: Structure, symbol attributes, storage and management.
Run-time environment: Procedure activation, parameter passing, value return, memory allocation, and scope. **[6hrs] (CO 2)**

UNIT 5: Intermediate code generation: intermediate representations, translation of declarations, assignments, control flow, Boolean expressions and procedure calls. Implementation issues.

[6hrs] (CO 3)

UNIT 6: Code generation and instruction selection: issues, basic blocks and flow graphs, register allocation, code generation, dag representation of programs, code generation from dags, peep hole optimization, Architecture dependent code improvement: instruction scheduling for pipeline, loop optimization for cache memory.

[8hrs] (CO 4, 5)

Course Outcomes:

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

CO 1: Understand the major phases of compilation including front-end and back-end.

CO 2: Develop the parsers and experiment the knowledge of different parsers design

CO 3: Construct the intermediate code representations and generation

CO 4: Convert source code for a novel language into machine code for a novel computer

CO 5: Apply for various optimization techniques for dataflow analysis

Text Books:

1. AlfredAho, Ravi Sethi, Jeffrey D Ullman, “Compilers Principles, Techniques and Tools”, Pearson Education Asia, 2003
2. Allen I. Holub “Compiler Design in C”, Prentice Hall of India, 2003.
3. C. N. Fischer and R. J. LeBlanc, “Crafting a compiler with C”, Benjamin Cummings, 2003.
4. J.P. Bennet, “Introduction to Compiler Techniques”, Second Edition, Tata McGraw-Hill, 2003.
5. HenkAlblas and Albert Nymeyer, “Practice and Principles of Compiler Building with C”, PHI, 2001.
6. Kenneth C. Louden, “Compiler Construction: Principles and Practice”, Thompson Learning, 2003

Course Code: BTCS602-18UC	Course Title : Artificial Intelligence	3L:1T:0P	3 Credits

Detailed Contents:

UNIT 1: Introduction: AI problems, Agents and Environments, Structure of Agents, Problem Solving Agents Basic Search Strategies: Problem Spaces, Uninformed Search (Breadth-First, Depth-First Search, Depth-first with Iterative Deepening), Heuristic Search (Hill Climbing, Generic Best-First, A*), Constraint Satisfaction (Backtracking, Local Search)

[**8hrs] (CO 1)**

UNIT 2: Advanced Search: Constructing Search Trees, Stochastic Search, A* Search Implementation, Minimax Search, Alpha-Beta Pruning Basic Knowledge Representation and Reasoning: Propositional Logic, First-Order Logic, Forward Chaining and Backward Chaining, Introduction to Probabilistic Reasoning, Bayes Theorem

[6hrs] (CO 2)

UNIT 3: Advanced Knowledge Representation and Reasoning: Knowledge Representation Issues, Nonmonotonic Reasoning, Other Knowledge Representation Schemes Reasoning Under Uncertainty: Basic probability, Acting Under Uncertainty, Bayes' Rule, Representing Knowledge in an Uncertain Domain, Bayesian Networks

[6hrs] (CO 3)

UNIT 4: Learning: What Is Learning? Rote Learning, Learning by Taking Advice, Learning in Problem Solving, Learning from Examples, Winston's Learning Program, Decision Trees.

[6hrs] (CO 4)

UNIT 5: Expert Systems: Representing and Using Domain Knowledge, Shell, Explanation, Knowledge Acquisition.

[6hrs] (CO 5)

Course Outcomes:

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

CO 1: Understand different types of AI agents.

CO 2: Develop different types of various AI search algorithms.

CO 3: Construct simple knowledge-based systems and to apply knowledge representation.

CO 4: Convert intermediate representation in contest to understand learning.

CO 5: Apply for various techniques for Expert Systems.

Text Book:

1. Russell, S. and Norvig, P, Artificial Intelligence: A Modern Approach, Third Edition, PrenticeHall, 2010.

Reference Books:

1. Artificial Intelligence, Elaine Rich, Kevin Knight, Shivasankar B. Nair, The McGraw Hill publications, Third Edition, 2009.
2. George F. Luger, Artificial Intelligence: Structures and Strategies for Complex Problem Solving, Pearson Education, 6th ed., 2009.

Course Code: BTCS606-18UC	Course Title: Network Security and Cryptography	3L:0T:0P	3Credits

Detailed Contents:

UNIT 1: **Overview of Network Security**
Internet Architecture Vulnerabilities, Network Security Terminology: Identification, Confidentiality, Authentication, Authorization, Access Control, Integrity, Non-Repudiation, Freshness, and Availability, Network Threats and Types of attacks, Introduction to malwares.

[4hrs] (CO 1,2)

UNIT 2: **Cryptography**
Symmetric Cipher Model, Classical Cryptographic Algorithms: Monoalphabetic Substitutions such as Caesar Cipher, Cryptanalysis of Monoalphabetic ciphers; Polyalphabetic Ciphers such as Vigenere, Vernam Cipher; Transposition Cipher. Stream and Block Ciphers, Block cipher: principles, Data Encryption Standard (DES), Analyzing and Strengthening of DES, Introduction to Advance Encryption Standard (AES), modes of operations, Concept of Asymmetric Cryptography, Rivets-Shamir-Adleman (RSA) Key Generation, Encryption and Decryption Algorithm

[6hrs]

UNIT 3: **Key Management Protocols:**
Solving Symmetric Key Distribution Problem, Diffie-Hellman Algorithm, Key Exchange with Public Key Cryptography or Asymmetric Cryptography, Digital Envelope, ELGamal Cryptosystem, Public Key Certificate Structure, Distribution of Public Key, Certificate Authority

[5hrs] (CO 3)

UNIT 4: **Hash Algorithms & Digital Signature**
Hash concept, Hash Function Requirements, Popular Message Digest and Hash Algorithms: MD4 and MD5, Secure Hash Algorithms such as SH1 and SHA2, Digital Signature, Digital Signature Standard (DSA)

[5hrs] (CO 2,4)

UNIT 5: **Authentication Protocols**
Basic authentication protocols, concept of Key distribution centre (KDC), Needham-Schroeder Authentication Protocol, Kerberos, writing authentication protocols using KDC and public key cryptography

[5hrs]

UNIT 6: **IP Security**
Why IP security: IP security Architecture, Authentication Header, Encapsulating Security Payload.

[5hrs] (CO 4)

UNIT 7: **Web Security**
Web security consideration, Secure Socket Layer Protocol, Transport Layer Security, Secure Electronic Transaction Protocol.

[4hrs] (CO 2,5)

UNIT 8: **Firewalls**
Firewall Design principles, Trusted Systems, Virtual Private Networks.

[4hrs] (CO 6)

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

CO 1: Understand real time systems for identifying security threats.
CO2: Compare public and private cryptographic algorithms and make use of the same for encryption and decryption of messages.
CO3: Design confidential systems with minimum possible threats.
CO4: Apply both cryptography and hashing to create digital signatures and certificates for achieving integrity.
CO5: Understand application of cryptosystems in design of, IPSec, AH, and ESP protocols.
CO6: Understand and compare https vs SET protocols and Firewall Vs Virtual Private Network.

Text Books:

1. Principles of Cryptography, 4th Edition by William Stallings, Pearson Education.
2. Security in Computing, 2nd Edition by Charles P.Pfleeger, Prentice Hall International.
3. Cryptography & Network Security, 2nd Edition by Atul Kahate, TMH.
4. Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2nd Edition by Bruce Schneier, John Wiley and Sons.
5. Firewalls and Internet Security, 2nd Edition by Bill Cheswick and Steve Bellovin, Addison-Wesley.
6. Security Technologies for the world wide web, 2nd Edition by Rolf Oppliger, Artech House, Inc.

Course Code: BTCS607-18UC	Course Title : Data Mining	3L: 0T: 0P	3Credits

Detailed Contents:

UNIT-1 **[8 hrs] (CO1)**

Introduction to data mining: Motivation and significance of data mining, data mining functionalities, interestingness measures, classification of data mining system, major issues in data mining, Knowledge Discovery in Databases Vs Data mining, DBMS Vs Data Mining, Data Mining Technique, DM Application Areas.

Data Ware Housing: Introduction, Multidimensional data model, OLAP Operation, Warehouse schema, Data Ware Housing Architecture, Warehouse Server, Metadata, OLAP, engine.

UNIT-II **[6 hrs] (CO2)**

Data pre-processing: Need, data summarization, data cleaning, data integration and transformation, data reduction techniques – Singular Value Decomposition (SVD), Discrete Fourier Transform (DFT), Discrete Wavelet Transform (DWT), data discretization and concept hierarchy generalization.

UNIT-III **[8 hrs] (CO3)**

Association rules: -Introduction, Methods to discover association rules, A Priori Algorithm, Partition Algorithm, Pincer –Search algorithm, Dynamic Item set counting algorithm, FP-tree Growth algorithm, Incremental algorithm, Border algorithm.

Clustering Techniques: - Introduction, Clustering paradigms, Partitioning algorithms, k-Mean Algorithm, k-Medoid Algorithm, CLARA, CLARANS, Hierarchical clustering, DBSCAN, BIRCH, CURE, Categorical clustering algorithms, STIRR, ROCK, CACTUS.

UNIT-IV **[6 hrs] (CO4)**

Classification and prediction: Definition, decision tree induction, Bayesian classification, rule-based classification, classification by backpropagation and support vector machines, associative classification, lazy learners, prediction, accuracy and error measures.

UNIT -V **[4 hrs] (CO5)**

Data mining on complex data and applications: Algorithms for mining of spatial data, multimedia data, text data; Data mining applications, social impacts of data mining, trends in data mining.

Course Outcomes:

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

CO1: Understand various concepts, algorithms and techniques in data mining and warehousing and their applications.

CO2: Develop preprocessing techniques for data cleansing

CO3: Construction algorithms for Association Rules and clustering techniques.

CO4: Identify different classification techniques.

CO5: Apply the real time applications of data mining

Text Books:

1. Han, J. and Kamber, M., "Data Mining - Concepts and Techniques", 3rd Ed., Morgan Kaufmann Series. 2011
2. Ali, A. B. M. S. and Wasimi, S. A., "Data Mining - Methods and Techniques", Cengage Publishers. 2009
3. Tan, P.N., Steinbach, M. and Kumar, V., "Introduction to Data Mining", Addison Wesley – Pearson. 2008
4. Pujari, A. K., "Data Mining Techniques", 4th Ed., Sangam Book 2008

Course Code: BTCS608-18UC	Course Title : Cloud Computing	3L:0T:1	3 Credits

Detailed Contents:

UNIT 1: **Introduction to Cloud Computing:** Origins of Cloud Computing– Cloud components - Essential characteristics – On-demand selfservice, Broad network access, Location independent resource pooling, Rapid elasticity, Measured service, Comparing cloud providers with traditional IT service providers, Roots of cloud computing. Cloud Computing Reference Model. Historical Developments. **Virtualization:** Introduction, Characteristics of Virtualized Environment, Taxonomy of Virtualization Techniques, Virtualization and Cloud computing, Pros and Cons of Virtualization, Technology Examples- Microsoft Hyper-V. **Before the Move into the Cloud**: Know Your Software Licenses, The Shift to a Cloud Cost Model, Service Levels for Cloud Applications.

[8hrs] (CO 1)

UNIT 2: Cloud Computing Architecture: Introduction, Architecture, Infrastructure / Hardware as a Service, Platform as a Service, Software as a Service, Types of Clouds, Public Clouds, Private Clouds, Hybrid Clouds, Community Clouds, Economics of the Cloud, Open Challenges, Cloud Interoperability and Standards, Scalability and Fault Tolerance. **Ready for the Cloud:** Web Application Design, Machine Image Design, Privacy Design, Database Management, Data Security, Network Security, Host Security, Compromise Response. Advantages of Cloud computing.

[6hrs] (CO 2)

UNIT 3: Defining the Clouds for Enterprise: Storage as a service, Database as a service, Process as a service, Information as a service, Integration as a service and Testing as a service. **Scaling a cloud infrastructure** - Capacity Planning, Cloud Scale. **Disaster Recovery:** Disaster Recovery Planning, Disasters in the Cloud, Disaster Management, Scheduling.

[6hrs] (CO 3)

UNIT 4: Cloud Simulators- CloudSim and GreenCloud : Introduction to Simulator, understanding CloudSim simulator, Understanding Working platform for CloudSim, Introduction to GreenCloud. **Introduction to VMWare Simulator:** Basics of VMWare, advantages of VMware virtualization, using Vmware workstation, creating virtual machines-understanding virtual machines, create a new virtual machine on local host, cloning virtual machines, virtualize a physical machine, starting and stopping a virtual machine**.**

[8hrs] (CO 4)

UNIT 5: Cloud Applications: Scientific Applications – Health care, Geoscience and Biology. Business and Consumer Applications- CRM and ERP, Social Networking, Media Applications and Multiplayer Online Gaming. **Cloud Platforms in Industry:** Amazon Web Services- Compute Services, Storage Services, Communication Services and Additional Services. Google AppEngine-Architecture and Core Concepts, Application Life-Cycle, cost model. Microsoft Azure- Azure Core Concepts, SQL Azure.

[6hrs] (CO 5)

Course Outcomes:

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

CO 1: Understand the necessary theoretical background for computing and storage clouds environments.

CO 2: Develop methodologies and technologies for the development of applications that will be deployed and offered through cloud computing environments.

CO 3: Construct the differences between Cloud deployment models

CO 4: Identify available Cloud Service Platforms and determine which works best for one's needs

CO 5: An understanding of when and where to use it using the appropriate industry models

Text Books:

1. Cloud Computing and SOA Convergence in Your Enterprise A Step-by-Step Guide by David S. Linthicum from Pearson 2010.
2. Cloud Computing 2 nd Edition by Dr. Kumar Saurabh from Wiley India 2012
3. Cloud computing for dummies- Judith Hurwitz , Robin Bloor , Marcia Kaufman ,Fern Halper, Wiley Publishing, Inc, 2010

Course Code: BTCS612-18UC	Course Title : Information Theory and Coding	3L:0T:0P	3Credits

Detailed Contents:

UNIT 1: Information Theory:

Introduction, Measure of information, Information content of message, Average Information content of symbols in Long Independent sequences, Average Information content of symbols in Long dependent sequences, Markov Statistical Model of Information Sources, Entropy and Information rate of Mark off Sources **[6hrs] (CO 1)**

UNIT 2: Source Coding:

Source coding theorem, Prefix Codes, Kraft McMillan Inequality property – KMI ,Encoding of the Source Output, Shannon's Encoding Algorithm , Shannon Fano Encoding Algorithm, Huffman codes, Extended Huffman coding, Arithmetic Coding, Lempel – Ziv Algorithm

[6hrs] (CO 2)

UNIT 3: Information Channels:

Communication Channels, Channel Models, Channel Matrix, Joint probability Matrix, Binary Symmetric Channel, System Entropies, Mutual Information, Channel Capacity, Channel Capacity of :Binary Symmetric Channel, Binary Erasure Channel, Moraga's Theorem, Continuous Channels

[6hrs] (CO 3)

UNIT4: Error Control Coding:
Introduction, Examples of Error control coding, methods of Controlling Errors, Types of Errors, types of Codes
Linear Block Codes: matrix description of Linear Block Codes, Error Detection and Error Correction Capabilities of Linear Block Codes, Single Error Correcting hamming Codes, Table lookup Decoding using Standard Array.
Binary Cyclic Codes: Algebraic Structure of Cyclic Codes, Encoding using an (n-k) Bit Shift register, Syndrome Calculation, Error Detection and Correction

[8hrs] (CO 4)

UNIT 5: Some Important Cyclic Codes: Golay Codes, BCH Codes , Convolution Codes: Convolution Encoder, Time domain approach, Transform domain approach, Code Tree, Trellis and State Diagram, The Viterbi Algorithm **[6hrs] (CO 5)**

Course Outcomes:

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

CO 1: Understand various entropies and Define the information theories.

CO 2: Apply source coding techniques

CO 3: Compute the capacity of various types of channels.

CO 4: Understand and Construct codes using different error control techniques.

CO 5: Apply various coding schemes for text, speech and audio.

Text Books:

1. Information Theory and Coding, Murlidhar Kulkarni and K.S.Shivaprakasha, Wiley India, 2014, EDN-1
2. Information Theory, Coding and Cryptography, Ranjan Bose, Mcgraw Hill, 2019, 3rd edition
3. Information Theory and Coding: Basics and Practices, V, Veluswamy, New Age International Pvt. ltd., 2014, 1st edition
4. Elements of Information Theory, 2nd Ed., T. M. Cover, J.A. Thomas, Wiley-Interscience, New York, 2006
5. Foundations of Coding: Theory and Applications of Error-Correcting Codes with an Introduction to Cryptography and Information Theory, J. Adamek, Wiley-Interscience 1991
6. Information Theory and Coding, Girithar K., Pooja publications, 2010

Course Code:BTCS613-18UC	Course Title: Data Science	3L:0T:0P	3 Credits

Detailed Contents: [**8hrs] (CO 1)**

UNIT1: **INTRODUCTION TO DATA SCIENCE**

Introduction to Data Science and Its Importance, Data Science and Life Cycle of Data Science, The Architecture of Data Science, Working with Data, Data Cleaning, Data Munging, Data Manipulation. Establishing Computational Environments for Data Scientists Using Python with IPython and Jupyter notebook

[6hrs] (CO 2)

UNIT2: DATA SCIENCE USING NUMPY

Understanding Data Types in Python, The Basics of NumPy, Usage of NumPy, Computation on Numpy Arrays, Usage Universal Functions - Aggregations: Min, Max, And Everything in Between Computation on Arrays, Broadcasting Comparisons, Masks, And Boolean Logic Fancy Indexing-Sorting Arrays

[6hrs] (CO 3)

UNIT 3: DATA MANIPULATION WITH PANDAS

Installing and Using Pandas, Introducing Pandas Objects, Data Indexing and Selection. Operating on Data in Pandas, Handling Missing Data, Hierarchical Indexing Combining Datasets: Concat And Append, Combining Datasets: Merge and Join. Aggregation and Grouping, Pivot Tables, Vectorized String Operations, Working with Time Series.

[6hrs] (CO 4)

UNIT 4: DATA VISUALIZATION WITH MATPLOTLIB

General Matplotlib Tips, Simple Line Plots, Simple Scatter Plots, Visualizing Errors Density and Contour Plots, Histograms, Binnings, And Density, Customizing Plot Legends Customizing Colour bars, Multiple Subplots, Text And Annotation, Customizing Ticks Customizing Matplotlib: Configurations And Style Sheets, Geographic Data With Base map.

[6hrs] (CO 5)

UNIT 5: MACHINE LEARNING USING PYTHON

Introduction of Machine Learning, Various Categories of Machine Learning algorithms, Archtecture of Machine Learning Algorithm, Basics of Supervised and Unsupervised Machine Learning Algorithm, Usage of Scikit Application, Feature Engineering- Naive Bayes Classification, Linear Regression, k-Means Clustering.

Course Outcomes:

CO 1: Identify phases involved in the life cycle of Data Science

CO2: Understanding the number of mathematical operations and computation on arrays

CO 3: Manage the data for efficient storage and manipulation in Python

CO 4: Explore a flexible range of data visualizations approaches in Python.

CO 5: Analyse and create a Machine Learning Model for various types of data.

Text Books:

1. Pragmatic Machine Learning with Python, Avishek Nag, BPB Publication April 2020
2. Data Science with Jupyter, Prateek Gupta, BPB Publication, January 2019
3. Python Data Science Handbook-Essential Tools for Working with Data, Jake Vander Plas, O'Reilly Media, 2016.
4. Data Science from Scratch: First Principles with Python, Joel Grus, O'Reilly, 2015.
5. Python for Data Analysis, Wes Mckinney, O'Reilly Media, 2013.
6. Fundamentals of Data Science, Samuel Burns, Amazon KDP printing and Publishing, 2019.

Course Code: BTCS614-18UC	Course Title: Soft Computing	3L:0T:0P	3Credits

Detailed Contents:

UNIT 1:

Introduction: What is Soft Computing? Difference between Hard and Soft computing, Requirement of Soft computing, Major Areas of Soft Computing, Applications of Soft Computing

[4hrs]

UNIT 2:

Neural Networks: What is Neural Network, Learning rules and various activation functions, Single layer Perceptrons, Back Propagation networks, Architecture of Backpropagation(BP) Networks, Backpropagation Learning, Variation of Standard Back propagation Neural Network, Introduction to Associative Memory, Adaptive Resonance theory and Self Organizing Map, Recent Applications.

[10hrs]

UNIT 3:

Fuzzy Systems: Fuzzy Set theory, Fuzzy versus Crisp set, Fuzzy Relation, Fuzzification, Minmax Composition, Defuzzification Method, Fuzzy Logic, Fuzzy Rule based systems, Predicate logic, Fuzzy Decision Making, Fuzzy Control Systems, Fuzzy Classification

[8 hrs]

UNIT 4:

Genetic Algorithm: History of Genetic Algorithms (GA), Working Principle, Various Encoding methods, Fitness function, GA Operators- Reproduction, Crossover, Mutation, Convergence of GA, Bit wise operation in GA, Multi-level Optimization

[8 hrs]

UNIT 5:

Hybrid Systems: Sequential Hybrid Systems, Auxiliary Hybrid Systems, Embedded Hybrid Systems, Neuro-Fuzzy Hybrid Systems, Neuro-Genetic Hybrid Systems, Fuzzy-Genetic Hybrid Systems

[5 hrs]

Course Outcomes:

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

- Understand various soft computing concepts for practical applications
- Design suitable neural network for real time problems
- Construct fuzzy rules and reasoning to develop decision making and expert system
- Apply the importance of optimization techniques and genetic programming
- Review the various hybrid soft computing techniques and apply in real time problems

Text Books:

1. S.Rajasekaran and G.A.Vijayalakshmi Pai, "Neural Networks, Fuzzy Logic and Genetic Algorithm: Synthesis and Applications", Prentice-Hall of India Pvt. Ltd., 2006.
2. George J. Klir, Ute St. Clair, Bo Yuan, Fuzzy Set Theory: Foundations and Applications Prentice Hall, 1997.
3. David E. Goldberg, Genetic Algorithm in Search Optimization and Machine Learning Pearson Education India, 2013.
4. James A. Freeman, David M. Skapura, Neural Networks Algorithms, Applications, and Programming Techniques, Pearson Education India, 1991.

Course Code: BTCS618-18UC	Course Title : Internet of Things	3L:0T:0P	3Credits

Detailed Contents:

1. **Introduction to IoT:**
   Architectural Overview, Design principles and needed capabilities, IoT Applications, Sensing, Actuation, Basics of Networking, M2M and IoT Technology FundamentalsDevices and gateways, Data management, Business processes in IoT, Everything as a Service(XaaS), Role of Cloud in IoT, Security aspects in IoT.

   (8 Hours), CO1

2. **Elements of IoT :**
   Hardware Components- Computing (Arduino, Raspberry Pi), Communication, Sensing, Actuation, I/O interfaces. Software Components- Programming API's (using Python /Node.js /Arduino) for Communication, Protocols-MQTT, ZigBee, Bluetooth, CoAP, UDP, TCP

   (9 Hours), CO2

3. **IoT Application Development**
   Solution framework for IoT applications- Implementation of Device integration, Data acquisition and integration, Device data storage- Unstructured data storage on cloud/local server, Authentication, authorization of devices.

   (18 Hours) CO3

4. **IoT Case Studies**
   IoT case studies and mini projects based on Industrial automation, Transportation, Agriculture, Healthcare, Home Automation, usage of Big Data Analytics as IoT, edge computing.

   (10 Hours), CO4

Course Outcomes: After the completion of this course, the students will be able to:
CO1: Understand internet of Things and its hardware and software components
CO2: Interface I/O devices, sensors & communication modules
CO3: Remotely monitor data and control devices
CO4: Develop real life IoT based projects

Text Books:

1. Vijay Madisetti, Arshdeep Bahga, Ïnternet of Things, "A Hands on Approach", University Press
2. Dr. SRN Reddy, Rachit Thukral and Manasi Mishra, "Introduction to Internet of Things: A practical Approach", ETI Labs
3. Pethuru Raj and Anupama C. Raman, "The Internet of Things: Enabling Technologies, Platforms, and Use Cases", CRC Press
4. Jeeva Jose, "Internet of Things", Khanna Publishing House, Delhi
5. Adrian McEwen, "Designing the Internet of Things", Wiley
6. Raj Kamal, "Internet of Things: Architecture and Design", McGraw Hill
7. Cuno Pfister, "Getting Started with the Internet of Things", O Reilly Media

---

Course Code: BTCS619-18UC	Course Title: Cyber Security	3L:0T:0P	3 Credits

Detailed Contents:

UNIT 1:
Introduction: Introduction to Computer Security, Threats, Harms, Vulnerabilities, Authentication, Access Control and Cryptography, Authentication, Access Control, Cryptography, Unintentional (Non-malicious) Programming Oversights, Malicious Code—Malware, Countermeasures

[8hrs]

UNIT 2:
Web Security: User Side, Browser Attacks, Web Attacks Targeting Users, Obtaining User or Website Data, Email Attacks

[6hrs]

UNIT 3:
Operating Systems Security: Security in Operating Systems, Security in the Design of Operating Systems, Rootkit

[6hrs]

UNIT 4:
Network and Cloud Security: Network Concepts, Threats to work Communications, Wireless Network Security, Denial of Service, Distributed Denial-of-Service Strategic Defenses: Security Countermeasures, Cryptography in Network Security, Firewalls, Intrusion Detection and Prevention Systems, Network Management, Cloud Computing Concepts, migrating to the Cloud, Cloud Security Tools and Techniques, Cloud Identity Management, Securing IaaS

[10hrs]

UNIT 5:
Privacy: Privacy Concepts, Privacy Principles and Policies, Authentication and Privacy, Privacy on the Web, Email Security
Management and Incidents: Security Planning, Business Continuity Planning, Handling Incidents, Risk Analysis, Dealing with Disaster
Legal Issues and Ethics: Protecting Programs and Data, Information and the Law, Rights of Employees and Employers, Issues related to Software Failures, Ethical Issues in Computer Security, Incident Analysis with Ethics

[10hrs]

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

CO 1: Understand the broad set of technical, social & political aspects of Cyber Security and security management methods to maintain security protection

CO2: Appreciate the vulnerabilities and threats posed by criminals, terrorist and nation states to national infrastructure

CO3: Understand the nature of secure software development and operating systems

CO4: Recognize the role security management plays in cyber security defense and legal and social issues at play in developing solutions.

CO5: Understand the concepts related to Security and Privacy.

Text Books:

1. Pfleeger, C.P., Security in Computing, Prentice Hall, 2010, 5th edition.

2. Schneier, Bruce. Applied Cryptography, Second Edition, John Wiley & Sons, 1996.

Reference Books:

1. Rhodes-Ousley, Mark. Information Security: The Complete Reference, Second Edition, Information Security Management: Concepts and Practice. New York, McGraw-Hill, 2013.
2. Whitman, Michael E. and Herbert J. Mattord. Roadmap to Information Security for IT and Infosec Managers. Boston, MA: Course Technology, 2011

Course Code:BTCS604-18UC	Course Title: Compiler Design Lab	0L:0T:2P	1 Credits

Detailed List of Tasks:

1. Write a program to design a Lexical Analyzer(LA) for a given language.
2. Write a program to implement the Lexical Analyzer using lex tool.
3. Write a program to design Predictive Parser (Non Recursive Descent parser) for a given language.
4. Write a program for constructing of LL (1) parsing.
5. Write a program for constructing recursive descent parsing.
6. Write a program to design and implement an LALR bottom up Parser for checking the syntax of the statements in a given language.
7. Write a program to implement operator precedence parsing.
8. Convert the BNF rules into Yacc form and write code to generate abstract syntax tree.
9. Write a program to generate machine code from the abstract syntax tree generated by the parser.
10. write a program to implement simple code optimization technique.
11. Write a program to generation of Code for a given Intermediate Code.

Suggested Tools – Lex, Yacc

Course Code: BTCS605-18UC	**Course Title: Artificial Intelligence Laboratory**	**0L:0T:2P**	**1 Credits**

Detailed List of Tasks:

1. Write a program for A* Algorithm.
2. Write a program for Depth First Search.
3. Write a program for Breadth First Search.
4. Write simple fact for the statements using PROLOG.
5. Write predicates one converts centigrade temperatures to Fahrenheit, the other checks if a temperature is below freezing.
6. Write a program in prolog for medical diagnosis and show the advantage and disadvantage of green and red cuts.
7. Write a program to implement tower of honoi.
8. Write a program to solve traveling salesman problem.
9. Write a program for expert system using forward chaining.

Course Code: BTCS609-18UC	**Course Title: Network Security and Cryptography Lab**	**0L:0T:2P**	**1Credits**

Detailed List of Tasks:

Task 1: What do you mean by a packet sniffer? What is its purpose? Explore from the Internet some popular packet sniffers. Wireshark is one such sniffer. Install Wireshark on your system. Explore its features and sniff various packets from your machine and enter into your machine. Identify the type of protocols of these packets. Connect to the PTU's website and find HTTP, TCP, IP and data link layer headers. Take snapshot of header fields, values and payloads of the packets being exchanged between your machine and PTU's website. **(CO 1)**

Task 2: Explore socket programming in C/python/java or any other technology/API for the purpose. Write a program to encrypt (using Ceaser cipher) your given plaintext into ciphertext at the client's machine and send the ciphertext using socket to server machine. Server should receive the ciphertext and transform it back to plaintext. Display plaintext at server's machine. **(CO 2)**

Task 3: Implement DES algorithm. Display all substitution and transposition outputs. **(CO 2)**

Task 4: Implement concept of digital envelop using socket programming. **(CO 2)**

Task 5: Write a program to implement RSA algorithm. **(CO 2)**

Task 6: Write a program to implement Diffie Hellman key exchange algorithm. Implement Man in the Middle attack. **(CO 2)**

Task 7: Explore various hash functions. Use these hash functions to generate digital signatures on different length messages. **(CO 3)**

Task 8: Design a secure message exchange system for PTU. Carefully identify the requirements and implement using socket programming. **(CO 3)**

Task 9: Install packet sniffer on your machine. Visit any https website. Take snapshots of TCP headers of all phases of SSL/TLS protocol. Demonstrate and explain working of SSL/TLS protocol with the help of snapshots. **(CO 1)**

Course Code: BTCS610-18UC	Course Title: Data Mining Lab	0L:0T:2P	1 Credits

Detailed List of Tasks:

1. Introduction to WEKA using JAVA.

2. To Perform data preprocessing tasks and Demonstrate performing association rule mining on data sets

3. Demonstrate performing clustering on data sets

4. Implementing SVM on real world problems.

5. Write a program to construct an optimized DECISION TREE for a given training data and by using any attribute selection measure.

6. Write a program for NAÏVE BAYESIAN algorithm for classifying the data.

7. Implement the K-Means Clustering algorithm for clustering the given data.

Suggested Tools- WEKA, Orange

Course Code: BTCS611-18UC	**Course Title: Cloud Computing Lab**	**0L:0T:2P**	**1 Credits**

Detailed List of Tasks:

1. Install Virtualbox/VMware Workstation with different flavours of linux or windows OS on top of windows7 or 8.
2. Install a C compiler in the virtual machine created using virtual box and execute Simple Programs
3. Installation of any open source tool. Create hello world app and other simple web applications using python/java.
4. Use of any open source tool for web applications using Saas.
5. Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.
6. Find a procedure to transfer the files from one virtual machine to another virtual machine.
7. Find a procedure to launch virtual machine using trystack (Online Openstack Demo Version)
8. Instructor may use CloudSim Architecture(User code, CloudSim, GridSim, SimJava)

Suggested Tools - CloudStack, FOSS-Cloud, Docker

Course Code: BTCS615-18UC	**Course Title: Information Theory and Coding Lab**	**0L:0T:2P**	**1 Credits**

Detailed List of Tasks:

1. Write a program for determination of various entropies and mutual information of a given channel. Test various types of channel such as a) Noise free channel. b) Error free channel c) Binary symmetric channel d) Noisy channel Compare channel capacity of above channels
2. Write a program for generation and evaluation of variable length source coding using C/MATLAB (Any 2) a) Shannon – Fanocodingand decoding b) Huffman Coding and decoding c) Lempel Ziv Coding and decoding
3. Write a Program for coding & decoding of Linear block codes
4. Write a Program for coding & decoding of Cyclic codes.
5. Write a program for coding and decoding of convolutional codes
6. Write a program for coding and decoding of BCH and RS codes
7. Write a simulation program to implement source coding and channel coding for transmitting a text file

Suggested Tools - Implementation can be done in C/C++/ MATLAB

Course Code: BTCS616-18UC	**Course Title: Data Science Lab**	**0L:0T:2P**	**1Credits**

Detailed List of Tasks:

1. Write a program for data preprocessing by using jupyter notebooks.
2. Write a program for data manipulation by using jupyter notebooks.
3. Write a program for counting a frequency of particular element in python.
4. Write any program for using numpy.
5. Write a program for using aggregation function.
6. Write a program for using sorting of an array using numpy.
7. Write any program for using pandas library.
8. Write any program for handling missing data using pandas library.
9. Write a program for concatenating and appending, combining datasets
10. Write a program for vectorizing string operations.
11. Write a basic program for using matplotlib library.
12. Write a program for simple line plots, simple scatter plots, visualizing errors density and contour plots by using matplotlib library.
13. Write a simple program for using scikit application.
14. Write a machine learning model that classify the data by using naive bayes classification
15. Write a machine learning model that classify the data by using linear regression.
16. Write a machine learning model that classify the data by using k-means clustering

Suggested Tools - Anaconda, PyCharm, Apache Spark, Spyder etc.

Course Code: BTCS617-18UC	Course Title: Soft Computing Lab	0L:0T:2P	1 Credits

Detailed List of Tasks:

1. Create a perceptron with appropriate no. of inputs and outputs. Train it using fixed increment learning algorithm until no change in weights is required. Output the final weights
2. Create a simple ADALINE network with appropriate no. of input and output nodes. Train it using delta learning rule until no change in weights is required. Output the final weights.
3. Train the auto correlator by given patterns: A1=(-1,1,-1,1), A2=(1,1,1,-1), A3=(-1, -1, - 1, 1). Test it using patterns: Ax=(-1,1,-1,1), Ay=(1,1,1,1), Az=(-1,-1,-1,-1).
4. Train the hetro correlator using multiple training encoding strategy for given patterns: A1=(000111001) B1=(010000111), A2=(111001110) B2=(100000001), A3=(110110101) B3(101001010). Test it using pattern A2.
5. Implement Union, Intersection, Complement and Difference operations on fuzzy sets. Also create fuzzy relation by Cartesian product of any two fuzzy sets and perform maxmin composition on any two fuzzy relations.
6. Solve Greg Viot's fuzzy cruise controller using MATLAB Fuzzy logic toolbox
7. Solve Air Conditioner Controller using MATLAB Fuzzy logic toolbox
8. Implement TSP using GA

Suggested Tools - MATLAB