

# **Scheme & Syllabus of** **Bachelor of Technology** **Computer Science & Engineering**

**Batch 2018 onwards**  
**(3<sup>rd</sup> -8<sup>th</sup> Semester)-**  
**Affiliated colleges**



By

Department of Academics  
**IK Gujral Punjab Technical**  
**University**

**IK Gujral Punjab Technical University, Kapurthala**  
**B. Tech, Computer Science & Engg.**

---

**Bachelor of Technology in Computer Science & Engineering**

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

**Courses & Examination**

**Scheme: Third Semester**

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

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

**IK Gujral Punjab Technical University,  
Kapurthala**

**Fourth Semester**

Course Code	Type of Course	Course Title	Hours per Week			Marks Distribution		Total Marks	Credits
			L	T	P	Internal	External		
<b>BTCS 401-18</b>	Professional Core Courses	Discrete Mathematics	3	1	0	40	60	100	4
<b>BTES 401-18</b>	Engineering Science Course	Computer Organization & Architecture	3	0	0	40	60	100	3
<b>BTCS 402-18</b>	Professional Core Courses	Operating Systems	3	0	0	40	60	100	3
<b>BTCS 403-18</b>	Professional Core Courses	Design & Analysis of Algorithms	3	0	0	40	60	100	3
<b>HSMC 122-18</b>	Humanities & Social Sciences including Management Courses	Universal Human Values 2	2	1	0	40	60	100	3
<b>EVS101-18</b>	Mandatory Courses	Environmental Sciences	3	-	-	100	-	100	S/US
<b>BTES 402-18</b>	Engineering Science Course	Computer Organization & Architecture Lab	0	0	2	30	20	50	1
<b>BTCS 404-18</b>	Professional Core Courses	Operating Systems Lab	0	0	4	30	20	50	2
<b>BTCS 405-18</b>	Professional Core Courses	Design & Analysis of Algorithms Lab	0	0	4	30	20	50	2
<b>Total</b>			<b>15</b>	<b>2</b>	<b>10</b>	<b>390</b>	<b>360</b>	<b>750</b>	<b>24</b>

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

**IK Gujral Punjab Technical University,  
Kapurthala**

**Fifth Semester**

Course Code	Type of Course	Course Title	Hours per Week			Marks Distribution		Total Marks	Credits
			L	T	P	Internal	External		
<b>BTES 501-18</b>	Engineering Science	Enterprise Resource Planning	3	0	0	40	60	100	3
<b>BTCS 501-18</b>	Professional Core Courses	Database Management Systems	3	0	0	40	60	100	3
<b>BTCS 502-18</b>	Professional Core Courses	Formal Language & Automata Theory	3	0	0	40	60	100	3
<b>BTCS 503-18</b>	Professional Core Courses	Software Engineering	3	0	0	40	60	100	3
<b>BTCS 504-18</b>	Professional Core Courses	Computer Networks	3	0	0	40	60	100	3
<b>BTCS XXX-18</b>	Professional Elective	Elective-I	3	0	0	40	60	100	3
<b>MC</b>	Mandatory Courses	Constitution of India/ Essence of Indian Traditional Knowledge	2	-	-	100	-	100	S/US
<b>BTCS 505-18</b>	Professional Core Courses	Database Management Systems Lab	0	0	4	30	20	50	2
<b>BTCS 506-18</b>	Professional Core Courses	Software Engineering Lab	0	0	2	30	20	50	1
<b>BTCS 507-18</b>	Professional Core Courses	Computer Networks Lab	0	0	2	30	20	50	1
<b>BTCS XXX-18</b>	Professional Elective	Elective-I Lab	0	0	2	30	20	50	1
	Professional Training	Industrial *Training	-	-	-	60	40	100	S/US
<b>Total</b>			<b>20</b>	<b>0</b>	<b>10</b>	<b>520</b>	<b>480</b>	<b>1000</b>	<b>23</b>

\* 4-6 weeks industrial training undertaken after 4<sup>th</sup> semester in summer vacations.

**IK Gujral Punjab Technical University,  
Kapurthala**

**Sixth Semester**

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

**IK Gujral Punjab Technical University,  
Kapurthala**

**Seventh Semester / Eighth Semester**

Course Code	Type of Course	Course Title	Hours per Week			Marks Distribution		Total Marks	Credits
			L	T	P	Internal	External		
<b>BTCS 701-18</b>	Professional Core Courses	Network Security and Cryptography	3	0	0	40	60	100	3
<b>BTCS 702-18</b>	Professional Core Courses	Data Mining and Data Warehousing	3	0	0	40	60	100	3
<b>BTOE ***</b>	Open Elective Courses	Open Elective-II	3	0	0	40	60	100	3
<b>BTCS ZZZ-18</b>	Professional Elective	Elective- IV	3	0	0	40	60	100	3
<b>BTCS TTT-18</b>	Professional Elective Courses	Elective-V	3	0	0	40	60	100	3
<b>BTCS 703-18</b>	Project	Project-II	0	0	12	120	80	200	6
<b>BTCS ZZZ-18</b>	Professional Elective	Elective- IV lab	0	0	2	30	20	50	1
<b>BTCS TTT-18</b>	Professional Elective	Elective- V lab	0	0	2	30	20	50	1
<b>Total</b>			<b>15</b>	<b>0</b>	<b>14</b>	<b>380</b>	<b>420</b>	<b>800</b>	<b>23</b>

**Seventh Semester / Eighth Semester**

Course Code	Course Title	Marks Distribution		Total Marks	Credits
		Internal	External		
<b>BTCS 801-18</b>	Semester Training	300	200	500	16

**LIST OF ELECTIVES**

**BTCS XXX-18: Elective-I**

<b>BTCS 510-18</b>	Programming in Python
<b>BTCS 513-18</b>	Programming in Python Lab
<b>BTCS 515-18</b>	Computer Graphics
<b>BTCS 518-18</b>	Computer Graphics lab
<b>BTCS 520-18</b>	Web Technologies
<b>BTCS 522-18</b>	Web Technologies lab
<b>BTCS 521-18</b>	Computational Biology
<b>BTCS 523-18</b>	Computational Biology lab

**BTCS UUU-18: Elective-II**

<b>BTCS 606-18</b>	Simulation and Modelling
<b>BTCS 607-18</b>	Simulation and Modelling Lab
<b>BTCS 608-18</b>	Internet of Things
<b>BTCS 609-18</b>	Internet of Things lab
<b>BTCS 610-18</b>	Digital Image processing
<b>BTCS 611-18</b>	Digital Image processing lab
<b>BTCS 612-18</b>	Cloud computing
<b>BTCS 613-18</b>	Cloud computing lab

**BTCS YYY-18: Elective-III**

<b>BTCS 614-18</b>	Software Project Management
<b>BTCS 615-18</b>	Software Project Management Lab
<b>BTCS 616-18</b>	Data Science
<b>BTCS 617-18</b>	Data Science lab
<b>BTCS 618-18</b>	Machine Learning
<b>BTCS 619-18</b>	Machine Learning lab
<b>BTCS 620-18</b>	Mobile Application Development
<b>BTCS 621-18</b>	Mobile Application Development lab

**BTCS ZZZ-18: Elective-IV**

<b>BTCS 704-18</b>	Deep Learning
<b>BTCS 705-18</b>	Deep Learning Lab
<b>BTCS 706-18</b>	Distributed databases
<b>BTCS 707-18</b>	Distributed databases lab
<b>BTCS 708-18</b>	Computer Vision
<b>BTCS 709-18</b>	Computer Vision lab
<b>BTCS 710-18</b>	Agile Software Development
<b>BTCS 711-18</b>	Agile Software Development lab

**BTCS TTT-18: Elective-V**

<b>BTCS 712-18</b>	Blockchain Technologies
<b>BTCS 713-18</b>	Blockchain Technologies Lab
<b>BTCS 714-18</b>	Parallel Computing
<b>BTCS 715-18</b>	Parallel Computing lab
<b>BTCS 716-18</b>	Adhoc and Wireless sensor networks
<b>BTCS 717-18</b>	Adhoc and Wireless sensor networks lab
<b>BTCS 718-18</b>	Quantum Computing
<b>BTCS 719-18</b>	Quantum Computing lab

**Open electives offered by the department:**

**BTCS301-18** Data Structures & Algorithms

**BTCS302-18** Object Oriented Programming

**BTES401-18** Computer organisation & Arcitecture

**BTCS402-18** Operating system

**BTCS501-18** Database Management System

**BTCS504-18** Computer Networks



**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

Course Code	Type of Course	Course Title	Hours per Week			Marks Distribution		Total Marks	Credits
			L	T	P	Internal	External		
BTCS30 1-18 & BTCS30 3-18	PCC	Data structure Theory & Lab	3	0	4	40T+30 P	60T+20 P	150	5
BTCS30 2-18 & BTCS30 4-18	PCC	Object Oriented Programming Theory & Lab	3	0	4	40T+30 P	60T+20 P	150	5
BTCS50 4-18 & BTCS50 7-18	PCC	Computer networks Theory & Lab	3	0	2	40T+30 P	60T+20 P	150	4
BTCS40 2-18 & BTCS40 4-18	PCC	Operating system Theory & Lab	3	0	4	40T+30 P	60T+20 P	150	5
BTES40 1-18 & BTCS40 2-18	ESC	Computer Organisation and architecture Theory & Lab	3	0	2	40T+30 P	60T+20 P	150	4
BTCS50 1-18 & BTCS50 4-18	PCC	Database Management system Theory & Lab	3	0	4	40T+30 P	60T+20 P	150	5

\*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

Course Code**	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+30P	60T+20P	150	4
	ELECTIVE	Machine Learning Theory & Lab	3	0	2	40T+30P	60T+20P	150	4
	ELECTIVE	Cloud computing Theory & Lab	3	0	2	40T+30P	60T+20P	150	4
	ELECTIVE	Adhoc and Sensor network Theory & Lab	3	0	2	40T+30P	60T+20P	150	4
	ELECTIVE	Data Science Theory & Lab	3	0	2	40T+30P	60T+20P	150	4
	ELECTIVE	Computer Graphics Theory & Lab	3	0	2	40T+30P	60T+20P	150	4
	ELECTIVE	Mobile Application Development Theory & Lab	3	0	2	40T+30P	60T+20P	150	4
	ELECTIVE	Data Mining & Warehousing Theory & Lab	3	0	2	40T+30P	60T+20P	150	4
	ELECTIVE	Information Theory & Coding Theory & Lab	3	0	2	40T+30P	60T+20P	150	4
	ELECTIVE	Soft Computing Theory & Lab	3	0	2	40T+30P	60T+20P	150	4

\*\* Refer to the scheme above for the course codes of respective courses.

# *Third Semester*

<b>Course Code:</b> BTCS301-18	<b>Course Title:</b> Data Structure & Algorithms	<b>3L:0T:P</b>	<b>3Credits</b>
--------------------------------	--	----------------	-----------------

**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, 2<sup>nd</sup> edition, PHI publishers.
2. “Fundamentals of Data Structures”, Illustrated Edition by Ellis Horowitz, Sartaj Sahni, 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.

---

<b>Course Code:</b> BTCS302-18	<b>Course Title:</b> Object Oriented Programming	<b>3L:0T:0P</b>	<b>3Credits</b>
--------------------------------	--	-----------------	-----------------

**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, Copyconstructors, 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**

**IK Gujral Punjab Technical University, Kapurthala**  
**B. Tech, Computer Science & Engg.**

Basics of exception handling, exception handling mechanism, throwing mechanism, catching mechanism, I/O System Basics, File I/O: 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, Josee Lajoie: C++ Primer, 4th Edition, Addison Wesley, 2012.
2. Herbert Schildt: The Complete Reference C++, 4th Edition, Tata McGraw Hill, 2011.

---

<b>Course Code:</b> BTCS303-18	<b>Course Title:</b> Data Structure & Algorithms Lab	<b>0L:0T:4P</b>	<b>2Credits</b>
--------------------------------	--	-----------------	-----------------

**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 array 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.

---

<b>Course Code:</b> BTCS304-18	<b>Course Title:</b> Object Oriented Programming Lab	<b>0L:0T:4P</b>	<b>2Credits</b>
--------------------------------	--	-----------------	-----------------

**List of Experiment:**

**IK Gujral Punjab Technical University, Kapurthala**  
**B. Tech, Computer Science & Engg.**

- 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.
-



**IK Gujral Punjab Technical University, Kapurthala**  
**B. Tech, Computer Science & Engg.**

<b>BTAM304-18</b>	<b>Mathematics Paper-III (Calculus and Ordinary Differential Equations)</b>	<b>4L:1T:0P</b>	<b>4 credits</b>
-------------------	---	-----------------	------------------

**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 most branches 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 solve engineering problems related to convergence, infinite series, power series and Taylor series;
4. Create, select and utilize the learnt techniques of first degree ordinary differential equations to model real world problems &;
5. Be acquainted with the knowledge required to solve higher order ordinary differential equations.

**Textbooks/References:**

1. G.B. Thomas and R.L. Finney, Calculus and Analytic geometry, 9<sup>th</sup> 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, 9<sup>th</sup> 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 Indian Philosophy.
- c. Greek Philosophy:

**2.2 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's Vakyapadiyam)
- 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)

**3. READINGS**

1. Copleston, Frederick, History of Philosophy, Vol. 1. Great Britain: Continuum.
2. Hiriyanna, M. Outlines of Indian Philosophy, Motilal Banarsidass Publishers; Fifth Reprint edition (2009)
3. Sathaye, Avinash, Translation of Nasadiya Sukta
4. Ralph T. H. Griffith. The Hymns of the R̥gveda. Motilal Banarsidass: Delhi: 1973.
5. Raju, P. T. Structural Depths of Indian Thought, Albany: State University of New York Press.
6. Plato, Symposium, Hamilton Press.
7. Kautilya Artha Sastra. Penguin Books, New Delhi.
8. Bacon, Nova Organum
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, Motilal Banarsidas, Delhi.
14. Passmore, John, Hundred Years of Philosophy, Penguin.

#### **4. OTHER SESSIONS:**

##### **4.1 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. Bhattacharyas, Daya Krishna, Gopinath Bhattacharya; comparative study of philosophical system such as Madhyastha Darshan.

#### **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.

<b>Course Code:BTES301-18</b>	<b>Course Title: Digital Electronics</b>	<b>3L:0T:0P</b>	<b>3Credits</b>
-------------------------------	--	-----------------	-----------------

## **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 and 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 Hill Publishing Company Limited, 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**, Pearson Education.
- 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*

<b>Course Code:</b> BTES401-18	<b>Course Title:</b> Computer Organization & Architecture	<b>3L:0T:0P</b>	<b>3Credits</b>
--------------------------------	---	-----------------	-----------------

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



**Reference Books:**

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

<b>Course Code:</b> BTCS402-18	<b>Course Title:</b> Operating Systems	<b>3L:0T:0P</b>	<b>3Credits</b>
--------------------------------	--	-----------------	-----------------

**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, Dining 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
-

<b>Course Code:</b> BTCS403-18	<b>Course Title:</b> Design and Analysis of Algorithms	<b>3L:0T:0P</b>	<b>3Credits</b>
--------------------------------	--	-----------------	-----------------

**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 write the 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, 4<sup>th</sup> edition, Pearson.
3. Fundamentals of Computer Algorithms – E. Horowitz, Sartaj Saini, Galgota Publications.

### **Reference Books**

1. Algorithm Design, 1<sup>st</sup> Edition, Jon Kleinberg and Éva Tardos, 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, Udi Manber, Addison-Wesley, Reading, MA.
- 

<b>Course Code:</b> BTES402-18	<b>Course Title:</b> Computer Organization & Architecture Lab	<b>0L:0T:2P</b>	<b>1Credits</b>
--------------------------------	---	-----------------	-----------------

### **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.
-

<b>Course Code:</b> BTCS404-18	<b>Course Title:</b> Operating Systems Lab	<b>0L:0T:4P</b>	<b>2Credits</b>
--------------------------------	--	-----------------	-----------------

### **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.
-

**List of Experiment:**

- Task 1:** Code and analyze solutions to following problem with given strategies:
- Knap Sack using greedy approach
  - 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:
- to find the topological sort of a directed acyclic graph
  - to find a path from source to goal in a maze.
- Task 5:** Implement an application of BFS such as:
- to find connected components of an undirected graph
  - 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 Floyd's algorithm.
- Task 9:** Code and analyze to find the minimum spanning tree in a weighted, undirected graph using Prim's algorithm.
- Task 10:** Code and analyze to find the minimum spanning tree in a weighted, undirected graph using Kruskal's 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, 4<sup>th</sup> edition, Pearson
2. Data Structures and Algorithms using Python and C++, David M. Reed and John Zelle, 2009 edition (available as e book), Franklin Beedle & Associates.

## **Course code: 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 Experiential Validation- as the process for self-exploration.
3. Continuous Happiness and Prosperity- A look at basic Human Aspirations
4. Right understanding, Relationship and Physical Facility- the basic requirements for 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 in harmony 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!**

7. Understanding human being as a co-existence of the sentient ‘I’ and the material ‘Body’
8. Understanding the needs of Self (‘I’) and ‘Body’ - happiness and physical facility
9. Understanding the Body as an instrument of ‘I’ (I being the doer, seer and enjoyer)
10. Understanding the characteristics and activities of ‘I’ and harmony in ‘I’
11. Understanding the harmony of I with the Body: Sanyam and Health; correct appraisal of Physical needs, meaning of Prosperity in detail
12. Programs to ensure Sanyam 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**

13. 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 of relationship.
14. Understanding the meaning of Trust; Difference between intention and competence
15. Understanding the meaning of Respect, Difference between respect and differentiation; the other salient values in relationship.
16. Understanding the harmony in the society (society being an extension of family): Resolution, Prosperity, fearlessness (trust) and co-existence as comprehensive Human Goals.
17. Visualizing a universal harmonious order in society- Undivided Society,

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**

18. Understanding the harmony in the Nature

19. Interconnectedness and mutual fulfilment among the four orders of nature - recyclability and self-regulation in nature

20. Understanding Existence as Co-existence of mutually interacting units in all-pervasive space

21. 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**

22. Natural acceptance of human values

23. Definitiveness of Ethical Human Conduct

24. Basis for Humanistic Education, Humanistic Constitution and Humanistic Universal Order

25. 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 appropriate technologies and management patterns for above production systems.

26. Case studies of typical holistic technologies, management models and production systems.

27. 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.

28. 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.

<b>Course Code: EVS101-18</b>	<b>Course Title: Environmental Studies-</b>	<b>L:2; T:0; P:0</b>	<b>0Credits</b>
-------------------------------	---	--------------------------	-----------------

.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 resources 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.
- India as a mega-diversity nation
- Hot-spots 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 rehabilitation 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
- vi) 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. Bharucha Erach, 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

<b>Course Code: HSMC101-18</b>	<b>Course Title: Development of Societies</b>	<b>3L:0T:0P</b>	<b>3Credits</b>
--------------------------------	---	-----------------	-----------------

### **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.

<b>Course Code: HSMC102-18</b>	<b>Course Title: PHILOSOPHY</b>	<b>3L:0T:0P</b>	<b>3Credits</b>
--------------------------------	---------------------------------	-----------------	-----------------

### **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's Vakyapadiyam)
- 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**

1. Copleston, Frederick, History of Philosophy, Vol. 1. Great Britain: Continuum.
2. Hiriyanna, M. Outlines of Indian Philosophy, Motilal Banarsidass Publishers; Fifth Reprint edition (2009)
3. Sathaye, Avinash, Translation of NasadiyaSukta
4. Ralph T. H. Griffith. The Hymns of the R̥gveda. Motilal Banarsidass: Delhi: 1973.
5. Raju, P. T. Structural Depths of Indian Thought, Albany: State University of New York Press.
6. Plato, Symposium, Hamilton Press.
7. Kautilya Artha Sastra. Penguin Books, New Delhi.
8. Bacon, Nova Organum
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, Motilal Banarsidas, 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. Bhattacharyas, 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 lived experience can be considered.

<b>BTCS401-18</b>	<b>Discrete Mathematics</b>	<b>3L:1T:0P</b>	<b>4 Credits</b>
-------------------	-----------------------------	-----------------	------------------

**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, Identities of Boolean Algebra, Duality, Representation of Boolean Function, Disjunctive and Conjunctive Normal Form CO4

**Module 5:**

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

**Suggested books:**

1. Kenneth H. Rosen, Discrete Mathematics and its Applications, Tata McGraw – Hill

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

**Suggested reference books:**

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

**Course Outcomes**

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

-----  
----

*Fifth  
Semester*

<b>BTES501-18</b>	<b>Enterprise Resource Planning</b>	<b>3L:0T:0P</b>	<b>3 Credits</b>
-------------------	-------------------------------------	-----------------	------------------

## **Course Details:**

### **UNIT 1 INTRODUCTION**

ERP: An Overview, Enterprise – An Overview, Benefits of ERP, ERP and Related Technologies, Business Process Reengineering (BPR), Data Warehousing, Data Mining, OLAP, SCM [9hrs., CO1]

### **UNIT II ERP IMPLEMENTATION**

ERP Implementation Lifecycle, Implementation Methodology, Hidden Costs, Organizing the Implementation, Vendors, Consultants and Users, Contracts with Vendors, Consultants and Employees, Project Management and Monitoring [9hrs., CO2]

### **UNIT III THE BUSINESS MODULES**

Business modules in an ERP Package, Finance, Manufacturing, Human Resources, Plant Maintenance, Materials Management, Quality Management, Sales and Distribution [9hrs., CO3]

### **UNIT IV THE ERP MARKET**

ERP Market Place, SAP AG, Peoplesoft, Baan, JD Edwards, Oracle, QAD, SSA [9hrs., CO4]

### **UNIT V ERP – PRESENT AND FUTURE**

Turbo Charge the ERP System, EIA, ERP and e-Commerce, ERP and Internet, Future Directions [6hrs., CO1]

## **TEXT BOOK**

1. Alexis Leon, “ERP Demystified”, Tata McGraw Hill, New Delhi, 2000

## **REFERENCES**

1. Joseph A Brady, Ellen F Monk, Bret Wagner, “Concepts in Enterprise Resource Planning”, Thompson Course Technology, USA, 2001.
2. Vinod Kumar Garg and Venkitakrishnan N K, “Enterprise Resource Planning – Concepts and Practice”, PHI, New Delhi, 2003

## **Course outcomes: The students at the end will be able;**

CO1: To know the basics of ERP

CO2: To understand the key implementation issues of ERP

CO3: To know the business modules of ERP

CO4: To be aware of some popular products in the area of ERP



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

### **Detailed Contents:**

#### **Module 1: Database system architecture**

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

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

#### **Module 2: Relational query languages**

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

**[10hrs] (CO2,4)**

#### **Module 3:**

Storage strategies, Indices, B-trees, hashing.

**[3hrs] (CO3)**

#### **Module 4: Transaction processing**

Concurrency control, ACID property, Serializability of scheduling, Locking and timestamp based schedulers, Multi-version and optimistic Concurrency Control schemes, Database recovery.

**[6hrs] (CO3)**

#### **Module 5: Database Security**

Authentication, Authorization and access control, DAC, MAC and RBAC models, Intrusion detection, SQL injection.

**[8hrs] (CO 4,5)**

#### **Module 6: Advanced Topics**

Object oriented and object relational databases, Logical databases, Web databases, Distributed databases.

**[8hrs] (CO 5)**

### **Course Outcomes:**

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

**CO1:** write relational algebra expressions for a query and optimize the Developed expressions

**CO2:** design the databases using ER method and normalization.

**CO3:** construct the SQL queries for Open source and Commercial DBMS-MYSQL, ORACLE, and DB2.

**CO4:** determine the transaction atomicity, consistency, isolation, and durability.

**CO5:** Implement the isolation property, including locking, time stamping based on concurrency control and Serializability of scheduling.

### **Text Books:**

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

### **Reference Books:**

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

<b>Course Code:</b> BTCS502-18	<b>Course Title:</b> Formal Language & Automata Theory	<b>3L:1T:0P</b>	<b>3Credits</b>	<b>42 Hours</b>
--------------------------------	--	-----------------	-----------------	-----------------

### **Detailed Contents**

#### **Module 1: Introduction**

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

#### **Module 2: Regular languages and finite automata:**

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

#### **Module 3: Context-free languages and pushdown automata**

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

#### **Module 4: Context-sensitive languages**

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

#### **Module 5: Turing machines**

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

#### **Module 6: Undecidability & Intractability:**

Church-Turing thesis, universal Turing machine, the universal and diagonalization languages, reduction between languages and Rice's theorem, undecidable problems about languages.

Intractability: Notion of tractability/feasibility. The classes NP and co-NP, their importance. Polynomial time many-one reduction. Completeness under this reduction. Cook-Levin theorem: NP-completeness of propositional satisfiability, other variants of satisfiability. NP-complete problems from other domains: graphs (clique, vertex cover, independent sets, Hamiltonian cycle), number problem (partition), set cover **[12hrs] (CO5 )**

**Course Outcomes:** The student will be able to:

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

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

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

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

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

**Text Books:**

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

**Reference Books:**

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

<b>Course Code:</b> BTCS503-18	<b>Course Title:</b> Software Engineering	<b>3L:1T:0P</b>	<b>3Credits</b>	<b>42 Hours</b>
--------------------------------	---	-----------------	-----------------	-----------------

**Detailed Contents:**

**Module 1:**

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

**[10hrs] (CO 1)**

**Module 2:**

Basic issues in software design, modularity, cohesion, coupling and layering, function-oriented software design: DFD and Structure chart, object modeling using UML, Object-oriented software development, user interface design. Coding standards and Code review techniques.

**[8hrs] (CO 2)**

**Module 3:**

Fundamentals of testing, White-box, and black-box testing, Test coverage analysis and test case design techniques, mutation testing, Static and dynamic analysis, Software reliability metrics, reliability growth modeling.

**[10hrs] (CO 3)**

**Module 4:**

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

#### **Module 5:**

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

[6hrs] (CO5)

#### **Text Books:**

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

#### **Reference Books:**

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

#### **Course Outcomes:**

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

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

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

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

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

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

-----

<b>Course Code:</b> BTCS 504-18	<b>Course Title:</b> Computer Networks	<b>3L:1T:0P</b>	<b>3Credits</b>	<b>42 Hours</b>
---------------------------------	--	-----------------	-----------------	-----------------

### **Detailed Contents:**

#### **Module 1: Data Communication Components**

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

**[8hrs] (CO1)**

#### **Module 2: Data Link Layer and Medium Access Sub Layer**

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

**[10 hrs] (CO2)**

#### **Module 3: Network Layer**

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

**[8 hrs] (CO3)**

#### **Module 4: Transport Layer**

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

**[8 hrs] (CO3)**

#### **Module 5: Application Layer**

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

**[8 hrs] (CO4)**

**Course Outcomes:** The student will be able to:

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

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

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

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

### **Text Books:**

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

### **Reference Books:**

1. Computer Networks, 8th Edition, Andrew S. Tanenbaum, Pearson New International Edition.

2. Internetworking with TCP/IP, Volume 1, 6th Edition Douglas Comer, Prentice Hall of India.
3. TCP/IP Illustrated, Volume 1, W. Richard Stevens, Addison-Wesley, United States of America.

-----

<b>Course Code:</b> BTCS505-18	<b>CourseTitle:</b> Database management System lab	<b>0L:0T:4P</b>	<b>2Credits</b>
--------------------------------	--	-----------------	-----------------

### **List of Experiments:**

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

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

### **Course Outcomes:**

- CO1:** This practical will enable students to retrieve data from relational databases using SQL.
- CO2:** students will be able to implement generation of tables using datatypes
- CO3:** Students will be able to design and execute the various data manipulation queries.
- CO4:** Students will also learn to execute triggers, cursors, stored procedures etc.

-----

<b>Course Code:</b> BTCS506-18	<b>Course Title:</b> Software Engineering Lab	<b>0L:0T:2P</b>	<b>1 Credits</b>
--------------------------------	---	-----------------	------------------

### **List of Experiments:**

- Task 1:** Study and usage of OpenProj or similar software to draft a project plan
- Task 2:** Study and usage of OpenProj or similar software to track the progress of a project
- Task 3:** Preparation of Software Requirement Specification Document, Design Documents and Testing Phase

**Task 4:** related documents for some problems

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

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

**Task 7:** To perform unit testing and integration testing

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

**Task 9:** Testing of a web site

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

-----

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

**List of Experiments:**

**Task 1:** To study the different types of Network cables and network topologies.

**Task 2:** Practically implement and test the cross-wired cable and straight through cable using clamping tool and network lab cable tester.

**Task 3:** Study and familiarization with various network devices.

**Task 4:** Familiarization with Packet Tracer Simulation tool/any other related tool.

**Task 5:** Study and Implementation of IP Addressing Schemes

**Task 6:** Creation of Simple Networking topologies using hubs and switches

**Task 7:** Simulation of web traffic in Packet Tracer

**Task 8:** Study and implementation of various router configuration commands

**Task 9:** Creation of Networks using routers.

**Task 10:**Configuring networks using the concept of subnetting

**Task 11:**Practical implementation of basic network command and Network configuration commands like ping, ipconfig, netstat, tracert etc. for troubleshooting network related problems.

**Task 12:**Configuration of networks using static and default routes.

**Course Outcomes:**

The students will be able to:

**CO1:** Know about the various networking devices, tools and also understand the implementation of network topologies;

**CO2:** Create various networking cables and know how to test these cables;

**CO3:** Create and configure networks in packet trace rtool using various network devices and topologies;

**CO4:** Understand IP addressing and configure networks using the subnet in;

**CO5:** Configure routers using various router configuration commands.

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

# ***ELECTIVES- I***



<b>Course Code:</b> BTCS 510-18	<b>Course Title:</b> Programming in Python	<b>3L:0T:0P</b>	<b>3 Credits</b>	<b>42 Hours</b>
---------------------------------	--	-----------------	------------------	-----------------

### **Detailed Contents:**

#### **Module 1:**

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

**[8hrs] (CO1)**

#### **Module 2:**

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

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

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

**[10hrs] (CO1,2)**

#### **Module 3:**

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

**[8hrs] (CO 2,3)**

#### **Module 4:**

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

WEB Programming: Introduction, Web Surfing with Python, Creating Simple Web Clients, Advanced Web Clients, CGI-Helping Servers Process Client Data, Building CGI Application Advanced CGI, Web (HTTP) Servers.

**[10hrs] (CO 4,6)**

#### **Module 5:**

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

**[6 hrs] (CO5)**

### **Text Books:**

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

### **Course Outcomes:**

The students should be able to:

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

**CO2:** Demonstrate proficiency in handling Strings and File Systems.

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

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

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

-----

<b>Course Code:</b> BTCS 513-18	<b>Course Title:</b> Programming in Python Lab	<b>0L:0T:2P</b>	<b>1 Credits</b>	<b>2 Hours/ week</b>
---------------------------------	--	-----------------	------------------	--------------------------

**Prerequisites:** Students should install Python.

**List of Experiments:**

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

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

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

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

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

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

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

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

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

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

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

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

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

**Task 13:** Write a program that accepts the lengths of three sides of a triangle as inputs. The program output should indicate whether or not the triangle is a right triangle (Recall from the Pythagorean Theorem that in a right triangle, the square of one side equals the sum of the squares of the other two sides).

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

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

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

- Task 17:** Write a program that inputs a text file. The program should print all of the unique words in the file in alphabetical order.
- Task 18:** Write a Python class to convert an integer to a roman numeral.
- Task 19:** Write a Python class to implement  $\text{pow}(x, n)$
- Task 20:** Write a Python class to reverse a string word by word.
- 

<b>cCourse Code:</b> BTCS521-	<b>Course Title:</b> Computational Biology	<b>3L:0T:0P</b>	<b>3 Credits</b>	<b>42 Hours</b>
-------------------------------	--	-----------------	------------------	-----------------

### **Detailed Contents:**

#### **Module 1: Introduction**

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

[10hrs] (CO)

#### **Module 2: More about RNA and DNA**

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

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

[10hrs] (CO)

#### **Module 3: Proteins**

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

[8hrs] (CO)

#### **Module 4: Computation and Biology**

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

[14hrs] (CO)

### **Course Outcomes:**

The student will be able to:

- CO1:** Understand the basic of cell structure, divisions involved in reproduction of a cell, and its generic functionality;
- CO2:** Recognize the base line elements of a RNA and DNA; including fundamental behind their complex structure;
- CO3:** Comprehend primary structure of the protein and various related data-sets.
- CO4:** Demonstrate the concept of gene sequence alignment and simulate various related algorithms for the same.

### **Text books**

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

### **Reference Books**

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

<b>Course Code:</b> BTCS523-18	<b>Course Title:</b> Computational Biology Lab	<b>0L:0T:2P</b>	<b>1 Credits</b>	<b>2 Hours/week</b>
--------------------------------	--	-----------------	------------------	---------------------

### **List of Experiments:**

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

**Task 2,3:** Parsing sequence file formats

- Sequences and Alphabets
- Sequences act like strings
- Slicing a sequence
- Turning Seq objects into strings
- Concatenating or adding sequences
- Changing case
- Nucleotide sequences and (reverse) complements
- Transcription
- Translation

**Task 4,5:** Sequence annotation objects

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

**Task 6,7,8:** BLAST

- Running BLAST over the Internet
- Running BLAST locally
  - Introduction
  - Standalone NCBI BLAST+
  - Other versions of BLAST

- Parsing BLAST output
- The BLAST record class
- Dealing with PSI-BLAST
- Dealing with RPS-BLAST

#### BLAST and other sequence search tools

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

#### **Task 9,10:** Multiple Sequence Alignment objects

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

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

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

#### **Quick Reference:**

<http://biopython.org/DIST/docs/tutorial/Tutorial.html#htoc106>

[https://biopython.readthedocs.io/en/latest/Tutorial/chapter\\_seq\\_objects.html](https://biopython.readthedocs.io/en/latest/Tutorial/chapter_seq_objects.html)

---

<b>Course Code:</b> BTCS 515-18	<b>Course Title:</b> Computer Graphics	<b>3L:0T:0P</b>	<b>3 Credits</b>	<b>45 Hours</b>
---------------------------------	--	-----------------	------------------	-----------------

## **Detailed Contents:**

### **Module 1:**

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

**[6hrs] (CO1)**

### **Module 2:**

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

**[6hrs] (CO1)**

### **Module 3:**

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

**[6hrs] (CO1,2)**

### **Module 4:**

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

**[6hrs](CO2)**

### **Module 5:**

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

**[6hrs] (CO2)**

### **Module 6:**

3 D Graphics and Visibility: Plane projections and its types, Vanishing points, Specification of a 3D view. Image and object precision, Hidden edge/surface removal or visible edge/surface determination techniques; z buffer algorithms, Depth sort algorithm, Scan line algorithm and Floating horizon technique.

**[6hrs] (CO2,3)**

### **Module 7:**

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

**[6hrs] (CO2,3)**

### **Module 8:**

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

**[3hrs] (CO3)**

## **Reference Books:**

1. D. Hearn and M.P. Baker, Computer Graphics: C version, 2nd Edition, PHI, 2004.
2. D.F. Rogers, Mathematical Elements for Graphics, 2nd Edition., McGraw Hill, 2004.

3. J.D. Foley et al, Computer Graphics, Principles and Practices, 2nd Edition, Addison Wasley, 2004.
4. Roy A. Plastock, Gordon Kalley, Computer Graphics, Schaum's Outline Series, 1986.

**Course Outcomes:** The students shall be able to:

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

CO2: Make the student present the content graphically.

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

---

<b>Course Code:</b> BTCS 518-18	<b>Course Title:</b> Computer Graphics Lab	<b>0L:0T:4P</b>	<b>2 Credits</b>	<b>2 Hours/week</b>
---------------------------------	--	-----------------	------------------	---------------------

**List of Experiments:**

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

<b>Course Code:</b> BTCS 520-18	<b>Course Title:</b> Web Technologies	<b>3L:0T:0P</b>	<b>3 Credits</b>	<b>42 Hours</b>
---------------------------------	---------------------------------------	-----------------	------------------	-----------------

**Detailed Contents:**

**Module 1:**

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

[6 hrs][CO1]

**Module 2:**

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

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

[7 hrs][CO2]

**Module 3:**

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

[7 hrs][CO3]

**Module 4:**

**Java Script:** Introduction, JavaScript's history and versions, Basic syntax, Variables, Data types, Statements, Operators, Functions, Arrays, Objects, dialog boxes, JavaScript DOM.

[7 hrs][CO4]

**Module 5:**

**PHP and MySQL:** Introduction and basic syntax of PHP, Data types, Variables, Decision and looping with examples, String, Functions, Array, Form processing, Cookies and Sessions, E-mail, PHP-MySQL: Connection to server.

[7 hrs][CO5]

**Module 6:**

**Ajax and JSON:** AJAX Introduction, AJAX Components, Handling Dynamic HTML with Ajax, Advantages & disadvantages, HTTP request, XMLHttpRequest Server Response.

JSON– Syntax, Schema, Data types, Objects, Reading and writing JSON on client and server. Using JSON in AJAX applications.

[8 hrs][CO6]

**Students shall be able to:**

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

**Text Books:**

1. Jeffrey C. Jackson, “Web Technologies: A Computer Science Perspective”, Pearson Education



2. Rajkamal, “Internet and Web Technology”, Tata McGraw Hill
3. Ray Rischpater, “JavaScript JSON Cookbook”, Packt Publishing.
4. Ivan Bayross, “Web Enabled Commercial Application Development using HTML, DHTML JavaScript, Perl, CGI”, BPB Publications.
5. Peter Moulding, “PHP Black Book”, Coriolis.

<b>Course Code:</b> BTCS 522-18	<b>Course Title:</b> Web Technologies Lab	<b>0L:0T:2P</b>	<b>1 credits</b>	<b>2 Hours/ week</b>
---------------------------------	---	-----------------	------------------	----------------------

### **List of Experiments:**

1. Configuration and administration Apache Web Server.
2. Develop an HTML page to demonstrate the use of basic HTML tags, Link to different HTML page and also link within a page, insertion of images and creation of tables.
3. Develop a registration form by using various form elements like input box, text area, radio buttons, check boxes etc.
4. Design an HTML page by using the concept of internal, inline, external style sheets.
5. Create an HTML file to implement the styles related to text, fonts, links using cascading style sheets
6. Create an HTML file to implement the concept of document object model using JavaScript
7. Create an HTML page including JavaScript that takes a given set of integer numbers and shows them after sorting in descending order.
8. Write an HTML page including any required JavaScript that takes a number from one text field in the range of 0 to 999 and shows it in another text field in words. If the number is out of range, it should show “out of range” and if it is not a number, it should show “not a number” message in the result box.
9. Create a PHP file to print any text using variable.
10. Demonstrate the use of Loops and arrays in PHP
11. Create a PHP file using GET and POST methods.
12. A simple calculator web application that takes two numbers and an operator (+, -, /, \* and %) from an HTML page and returns the result page with the operation performed on the operands.
13. Implement login page contains the user name and the password of the user to authenticate with Session using PHP and MySQL, also implement this with the help of PHP-Ajax.
14. A web application for implementation:
  - a. The user is first served a login page which takes user’s name and password. After submitting the details the server checks these values against the data from a database and takes the following decisions.
  - b. If name and password matches, serves a welcome page with user’s full name.
  - c. If name matches and password doesn’t match, then serves “password mismatch” page
  - d. If name is not found in the database, serves a registration page, where user’s full name is asked and on submitting the full name, it stores, the login name, password and full name in the database (hint: use session for storing the submitted login name and password)
15. Demonstrate the use of Ajax and JSON Technologies in programming examples.
16. Demonstrate the use of web site designing tools such as Joomla, WordPress.
17. Implement at least one minor project using different technologies mentioned in theory of the subject.

*Sixth Semester*

<b>Course Code: BTCS601-18</b>	<b>Course Title : Compiler Design</b>	<b>3L:0T:0P</b>	<b>3Credits</b>
--------------------------------	---------------------------------------	-----------------	-----------------

### **Detailed Contents:**

#### **UNIT 1:** Unit I Introduction to Compilers:

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

#### **Unit II :**Syntax Analysis:

Role of Parser – Grammars – Error Handling – Context-free grammars – Writing a grammar, Top-Down Parsing – General Strategies Recursive Descent Parser – Predictive Parser-LL(1) Parser-Shift Reduce Parser-LR Parser-LR (0) Item Construction of SLR Parsing Table - Introduction to LALR Parser – Error Handling and Recovery in Syntax Analyzer-YACC. **[8 hrs., CO 2]**

#### **Unit III :** Intermediate Code Generation:

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

#### **Unit IV:** Run-Time Environment and Code Generation:

Storage Organization, Stack Allocation Space, Access to Non-local Data on the Stack, Heap Management – Issues in Code Generation – Design of a simple Code Generator. **[6 hrs., CO 4]**

#### **Unit V:** Code Optimization:

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

### **Course Outcomes:**

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

- CO1: Build concepts on lexical analysis.
- CO2: Understand strategies of syntax analysis.
- CO3: Learn techniques of Intermediate code generation.
- CO4: Understand code design issues and design code generator.
- CO5: Design and develop optimized codes.

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

1. A.V. Aho, Monica, R.Sethi, J.D.Ullman, “Compilers, Principles, Techniques and Tools”, Second Edition, Pearson Education/Addison Wesley, 2009.

2. Andrew W. Appel, “Modern Compiler Implementation in Java”, Second Edition, 2009.
3. J.P. Tremblay and P.G. Sorrenson, “The Theory and Practice of Compiler Writing”, McGraw Hill, 1985.

<b>Course Code:</b> <b>BTCS604-18</b>	<b>Course Title: Compiler Design Lab</b>	<b>L:0;T:0; 2P</b>	<b>1Credits</b>
--	--	--------------------	-----------------

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

<b>Course Code:BTCS602-18</b>	<b>Course Title : Artificial Intelligence</b>	<b>3L:0T:0P</b>	<b>3Credits</b>
-------------------------------	---	-----------------	-----------------

### **Detailed Contents:**

#### **UNIT 1: Introduction (3 Hours)**

Concept of AI, history, current status, scope, agents, environments, Problem Formulations, Review  
of tree and graph structures, State space representation, Search graph and Search tree.

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

#### **UNIT 2: Search Algorithms**

Random search, Search with closed and open list, Depth first and Breadth first search, Heuristic search, Best first search, A\* algorithm, Game Search.

**[9hrs] (CO 2)**

#### **UNIT 3: Probabilistic Reasoning**

Probability, conditional probability, Bayes Rule, Bayesian Networks- representation, construction and inference, temporal model, hidden Markov model.

**[6hrs] (CO 3)**

**UNIT 4 Markov Decision process**

MDP formulation, utility theory, utility functions, value iteration, policy iteration and partially observable MDPs.

**[6hrs] (CO 4)**

**UNIT 5 Reinforcement Learning**

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

**[6hrs] (CO 5)**

**Course Outcomes:**

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

CO1: Build intelligent agents for search and games

CO2: Solve AI problems by learning various algorithms and strategies

CO3: Understand probability as a tool to handle uncertainty

CO4: Learning optimization and inference algorithms for model learning

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

**Suggested Readings/ Books:**

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

<b>Course Code:</b> <b>BTCS 605-18</b>	<b>Course Title Artificial Intelligence Lab</b>	<b>L:0;T:0;2</b> <b>P:</b>	<b>1 Credits</b>
---	---	-------------------------------	------------------

**Detailed List of Tasks:**

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

# ELECTIVE II

<b>Course Code:</b> BTCS 606-18	<b>Course Title:</b> Simulation and Modeling	<b>3L:0T:0P</b>	<b>3Credits</b>
---------------------------------	--	-----------------	-----------------

### **Detailed Contents:**

#### **UNIT1:** Introduction

Introduction to simulation and modeling, Application areas, System and system environment, Components of a system, Discrete and continuous systems, Basic model forms and its types, Discrete-event simulation, Steps in a simulation study, Simulation examples. **[4 hrs] (CO 1)**

#### **UNIT2:** General Principles

Concepts in discrete event simulation, Handling Stepped and Event-based Time in Simulations, Event scheduling/time advance algorithms, World views, List processing using dynamic allocation and linked list. **[4 hrs] (CO 1)**

#### **UNIT 3:** Statistical and Queuing Models in Simulation

Terms and concepts, Statistical models, Discrete and continuous distributions, Poisson distributions, Empirical distributions, Little's equation. Characteristics of queuing systems, Queuing notation, Long- Run measures of performance of queuing systems, Steady state behavior of infinite and finite calling population models, Use of network of queues. **[9 hrs] (CO 2)**

#### **UNIT 4** Random Number Generation

Pseudo random numbers, Techniques for generation of pseudo random numbers, Tests for random numbers, Random variate generation, Inverse Transform Technique- Exponential, Uniform, Weibull, Triangular distributions, Direct transformation for Normal and lognormal distributions. **[6hrs] (CO 2)**

#### **UNIT 5** Input Modeling and Output Analysis of a Single Model

Data collection, Identifying the distribution of data - histograms and quantile plots, Parameter estimation, Goodness of fit tests applied to simulation inputs, Verification and validation of simulation models, Output analysis and measures of performance and estimation. **[6hrs] (CO 3)**

#### **UNIT 6** Comparison and Evaluation of Alternative System Designs

Comparison of two system designs, Sampling with equal and unequal variances, Common random numbers, Comparison of several system designs, Linear regression, Random number assignment for regression. **[5 hrs] (CO 4)**

### **Course Outcomes:**

After undergoing this course, the students will be able to

CO1: Discuss the fundamental elements of discrete-event simulation including statistical models, random processes, random variates, and inputs to simulation

CO2: Analyze a real world problem and apply modelling methodologies to develop a discrete-event simulation model

CO3 Interpret discrete-event techniques for solving a simulation problem

CO4: Compare and evaluate alternative system designs using sampling and regression

**Suggested Readings/ Books:**

1. Jerry Banks, John S. Carson II, Barry L.Nelson and David M.Nicol, "Discrete- event system and simulation", Prentice Hall of India.
2. Averill M.Law, "Simulation modeling and analysis (SIE)", Tata McGraw Hill India.
3. David Cloud, Larry Rainey, "Applied Modeling and Simulation", Tata McGraw Hill.
4. Gabriel A. Wainer, "Discrete-event modeling and simulation: a practitioner's approach", CRC Press.
5. Bernard P. Zeiger, Herbert Praehofer, Tag Gon Kim, "Theory of modeling and simulation: integrating discrete event and continuous complex dynamic systems", Academic Press.
6. Walter J. Karplus, George A. Bekey, Boris YakobKogan, "Modeling and simulation: theory and practice", Springer.

-----  
-----

<b>CourseCode:</b> BTCS 607-18	<b>CourseTitle:</b> Simulation and Modeling Lab	<b>L:0;T:0; P:</b>	<b>Credits</b>
--------------------------------	---	--------------------	----------------

**Detailed List of Tasks:**

1. Implementation of Basic Operations on Matrices.
2. Implementation of Chi-square goodness-of-fit test.
3. Practical implementation of Queuing Models.
4. Design Inventory System.
5. Implementation of Monte-Carlo Simulation method.
6. Analysis of Discrete and Continuous Distributions.
7. Generation of Random Numbers using Linear Congruential Method.
8. Generation of Random Numbers using Combined Linear Congruential Method.
9. Evaluation of system design using Regression Analysis.
10. Simulate a network using any network simulator.

**SuggestedTools** - Scilab, Tortuga and Extend. Introduction to network simulators - NS2, CloudSim, Wireshark.

<b>Course Code:</b> BTCS608-18	<b>Course Title:</b> Internet of Things	<b>L:3; T:0; P:0</b>	<b>3Credits</b>
--------------------------------	---	----------------------	-----------------

**DETAIL CONTENTS**

**1. Introduction to IoT**

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.

**(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

**(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

#### **List of suggested books :**

1. Vijay Madiseti, Arshdeep Bahga, 'Internet 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

<b>Course Code: BTCS609-18</b>	<b>Course Title: Internet of Things Lab</b>	<b>L:0; T:0; P:2</b>	<b>1Credits</b>
--------------------------------	---	--------------------------	-----------------

#### **LIST OF PRACTICALS**

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 thingspeak cloud.
10. Write a program on Arduino/Raspberry Pi to retrieve temperature and humidity data from thingspeak 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.

<b>Course Code:BTCS 610-18</b>	<b>Course Title : Digital Image Processing</b>	<b>3L:0T:0P</b>	<b>3Credits</b>
--------------------------------	--	-----------------	-----------------

### **Detailed Contents:**

#### **UNIT 1: Introduction of Digital Image Processing (DIP)**

Introduction to the DIP areas and applications; Components of Digital Image Processing; Elements of Visual Perception; Image Sensing and Acquisition; Image Sampling and Quantization; Relationships between pixels; color models.

[7hrs] (CO 1)

#### **UNIT 2: Image Enhancement**

**Spatial Domain:** Gray level transformations; Histogram processing; Basics of Spatial Filtering; Smoothing and Sharpening Spatial Filtering

**Frequency Domain:** Introduction to Fourier Transform; Smoothing and Sharpening frequency domain filters; Ideal, Butterworth and Gaussian filters

[10hrs] (CO 2)

#### **UNIT 3: Image Restoration**

Noise models; Mean Filters; Order Statistics; Adaptive filters; Band reject Filters; Band pass Filters; Notch Filters; Optimum Notch Filtering; Inverse Filtering; Wiener filtering

[8hrs] (CO 3)

#### **UNIT4: Feature Extraction and Image Segmentation**

**Feature Extraction:** Contour and shape dependent feature extraction, Extraction of textural features

**Segmentation:** Detection of Discontinuities; Edge Linking and Boundary detection; Region based segmentation; Morphological processing- erosion and dilation.

[10hrs] (CO 4)

#### **UNIT 5: Image Compression and Encoding**

Entropy-based schemes, Transform-based encoding, Predictive encoding and DPCM, Vector quantization, Huffman coding.

[10hrs](CO 5)

#### **Course Outcomes:**

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

CO1: Understand the basic concepts of DIP.

CO2: Improve the quality of digital images.

CO3: Understand and De-noise Digital Images

CO4: Segment digital images and extract various features from digital images

CO5: Understand various image compression techniques and apply such techniques to compress digital images for reducing the sizes of digital images.

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

1. Rafael C. Gonzales, Richard E. Woods, "Digital Image Processing", Third Edition, Pearson Education, 2010.
  2. Anil Jain K. "Fundamentals of Digital Image Processing", PHI Learning Pvt. Ltd., 2011.
  3. William K Pratt, "Digital Image Processing", John Wiley, 2002.
  4. Nick Efford, "Digital Image Processing a practical introduction using Java", Third Edition, Pearson Education, 2004.
  5. R.C. Gonzalez, R.E. Woods, and S. L. Eddins "Digital Image Processing using MATLAB", Pearson Prentice-Hall, 2004.
  6. Sandipan Dey, "Hands-On Image Processing with Python", Packt, 2018
- 

<b>Course Code:</b> BTCS 611-18	<b>Course Title: Digital Image Processing Lab</b>	<b>L:0;T:0; P:</b>	<b>Credits</b>
---------------------------------	---	--------------------	----------------

#### **Detailed List of Tasks:**

1. WAP to draw Histogram of digital Image
2. WAP to enhance the quality of digital image using various gray level transformations.
3. WAP to enhance the quality of digital image using Average and median filters in spatial domain.
4. WAP to convert digital image from spatial domain to frequency domain.
5. Implement low pass filters in frequency domain for image enhancement.
6. Implement high pass filters in frequency domain for image enhancement.
7. Implement Optimum Notch Filtering for de-noising of digital image.
8. WAP to segment digital image using thresholding approach.
9. WAP to extract shape and texture based features from image.
10. WAP to compress digital image using entropy based approach.

**Suggested Tools – MATLAB/Python/JAVA**

---

<b>Course Code: BTCS 612-18</b>	<b>Course Title: Cloud Computing</b>	<b>3L:0T:0P</b>	<b>3Credits</b>
---------------------------------	--------------------------------------	-----------------	-----------------

### Detailed Contents:

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

**[8hrs] (CO1)**

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

**[9hrs] (CO2)**

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

**[6hrs] (CO3)**

**UNIT 4: Cloud deployment models:** Introduction to cloud deployment models, Public clouds, Private clouds, Hybrid clouds, Community clouds, Migration paths for cloud, Selection criteria for cloud deployment.

**[6hrs] (CO4)**

**UNIT 5: Security in cloud computing:** Understanding security risks, Principal security dangers to cloud computing, Internal security breaches, User account and service hijacking, measures to reduce cloud security breaches

Case Studies: Comparison of existing Cloud platforms /Web Services.

**[6hrs] (CO5)**

### Course Outcomes:

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

CO1: Understand the core concepts of the cloud computing paradigm

CO2: Understanding importance of virtualization along with their technologies

CO3: Analyze various cloud computing service and deployment models and apply them to solve problems on the cloud.

CO4: Implementation of various security strategies for different cloud platform

### Suggested Readings/ Books:

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

### Reference Books

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

<b>Course Code: BTCS 612-18</b>	<b>Course Title: Cloud Computing Lab</b>	<b>L:0;T:0; P:2</b>	<b>1 Credits</b>
---------------------------------	--	---------------------	------------------

#### **Detailed List of Tasks:**

11. Install VirtualBox/VMware Workstation on different OS.
12. Install different operating systems in VMware.
13. Simulate a cloud scenario using simulator.
14. Implement scheduling algorithms.
15. To study cloud security management.
16. To study and implementation of identity management
17. Case Study - Amazon Web Services/Microsoft Azure/Google cloud services.

#### **Suggested Tools –Matlab, Cloudsim**

---

<b>Course Code: BTCS 616-18</b>	<b>Course Title : Data Science</b>	<b>3L:0T:0P</b>	<b>3Credits</b>
---------------------------------	------------------------------------	-----------------	-----------------

### **Detailed Contents:**

#### **UNIT 1: Introduction**

Introduction to Data Science , Evolution of Data Science , Data Science Roles , Stages in a Data Science Project, Applications of Data Science in various fields, Data Security Issues.

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

#### **UNIT 2: Data Collection and Data Pre-Processing**

Data Collection Strategies, Data Pre-Processing Overview, Data Cleaning, Data Integration and Transformation, Data Reduction, Data Discretization.

**[9hrs] (CO 2)**

#### **UNIT 3: Exploratory Data Analytics**

Descriptive Statistics, Mean, Standard Deviation, Skewness and Kurtosis , Box Plots, Pivot Table, Heat Map, Correlation Statistics , ANOVA.

**[6hrs] (CO 2)**

#### **UNIT 4: Model Development**

Simple and Multiple Regression, Model Evaluation using Visualization , Residual Plot, Distribution Plot, Polynomial Regression and Pipelines, Measures for In-sample Evaluation, Prediction and Decision Making.

**[6hrs] (CO 3)**

#### **UNIT 5 Model Evaluation**

Generalization Error, Out-of-Sample Evaluation Metrics, Cross Validation, Overfitting, Under Fitting and Model Selection, Prediction by using Ridge Regression, Testing Multiple Parameters by using Grid Search.

**[6hrs] (CO**

**4)**

### **Course Outcomes:**

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

CO1: understand of the basics of the ethical use of data science

CO2: Build skills in transformation and merging of data for use in analytic tools

CO3: Perform linear and multiple linear regression analyses.

CO4: Evaluate outcomes and make decisions based on data

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

1. Jojo Moolayil, “Smarter Decisions : The Intersection of IoT and Data Science”, PACKT, 2016.
2. Cathy O’Neil and Rachel Schutt , “Doing Data Science”, O’Reilly, 2015.

3. David Dietrich, Barry Heller, Beibei Yang, “Data Science and Big data Analytics”, EMC 2013.
4. Raj, Pethuru, “Handbook of Research on Cloud Infrastructures for Big Data Analytics”, IGI Global

-----  
-----

<b>Course Code: BTCS 617-18</b>	<b>Course Title : Data Science Lab</b>	<b>0L:0T:2P</b>	<b>1Credits</b>
---------------------------------	--	-----------------	-----------------

**Detailed List of Tasks:**

1. Creating and displaying Data. and Matrix manipulations
2. Creating and manipulating a List and an Array
3. Creating a Data Frame and Matrix-like Operations on a Data Frame
4. Merging two Data Frames and Applying functions to Data Frames
5. Visualization Effects
6. Plotting with Layers
7. Overriding Aesthetics
8. Histograms and Density Charts
9. Simple Linear Regression – Fitting, Evaluation and Visualization
10. Multiple Linear Regression, Lasso and Ridge Regression

**Suggested Tools – Python, R**

---

# ELECTIVE III



<b>Course Code: BTCS 614-18</b>	<b>Course Title: Software Project Management</b>	<b>3L:0T:0P</b>	<b>3 Credits</b>
---------------------------------	--	-----------------	------------------

### **Detailed Contents:**

#### **MODULE 1: Introduction**

Project Evaluation and Planning - Activities in Software Project Management, Overview of Project Planning, Stepwise planning, contract management, Software processes and process models.

**[5hrs] (CO1)**

#### **MODULE 2: Cost Benefit Analysis**

Cost Benefit Analysis, Cash Flow Forecasting, Cost-Benefit Evaluation Techniques, Risk Evaluation. Project costing, COCOMO 2, Staffing pattern, Effect of schedule compression, Putnam's equation, Capers Jones estimating rules of thumb.

**[6hrs] (CO2)**

#### **MODULE 3: Project Scheduling**

Project Sequencing and Scheduling Activities, Scheduling resources, Critical path analysis, Network Planning, Risk Management, Nature and Types of Risks, Managing Risks, Hazard Identification, Hazard Analysis, Risk Planning and Control, PERT and Monte Carlo Simulation techniques.

**[8hrs] (CO3)**

#### **MODULE 4: Monitoring & Control**

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

**[8hrs] (CO4)**

#### **MODULE 5: Quality Management**

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

**[9hrs] (CO5)**

### Course Outcomes:

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

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

### Suggested Readings/Books:

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

Course Code: BTCS 615-18	Course Title: Software Project Management Lab	L:0;T:0; P:2	1 Credits
--------------------------	---	--------------	-----------

### Detailed List of Tasks:

**Task 1:** Introduction to MS Project

**Task 2:** Create a Project Plan

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

**Task 3:** Create Project Plan contd.

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

**Task 4: Execute and Monitor the Project Plan**

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

**Task 5: Generate Dashboard and Reports**

- Dashboard
- Resource Reports
- Cost Reports
- Progress Reports

**Suggested Tools** – MS Project, Rational Team Concert

**Course Outcomes:**

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

CO1: Plan and manage projects.

CO2: Consolidate and communicate information about their project.

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

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

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

<b>Course Code: BTCS 618-18</b>	<b>Course Title : Machine Learning</b>	<b>3L:0T:0P</b>	<b>3Credits</b>
---------------------------------	--	-----------------	-----------------

**Detailed Contents:**

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

**[4hrs] (CO 1)**

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

**[4hrs] (CO 2)**

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

**[6hrs] (CO 3)**

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

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

### Course Outcomes:

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

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

CO2: Analyse and extract features of complex datasets

CO3: Deploy techniques to comment for the Regression

CO4: Comprehend and apply different classification and clustering techniques

CO5: Understand the concept of Neural Networks and Genetic Algorithm

### Suggested Readings/ Books:

Text Books:

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

Reference Books:

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

<b>Course Code: BTCS619-18</b>	<b>Course Title: machine Learning Lab</b>	<b>L:0;T:0;2 P:</b>	<b>1Credits</b>
--------------------------------	---	-------------------------	-----------------

### Detailed List of Tasks:

1. Implement data pre-processing
2. Deploy Simple Linear Regression
3. Simulate Multiple Linear Regression
4. Implement Decision Tree

5. Deploy Random forest classification
6. Simulate Naïve Bayes algorithm
7. Implement K-Nearest Neighbors (K-NN), k-Means
8. Deploy Support Vector Machine, Apriori algorithm
9. Simulate Artificial Neural Network
10. Implement the Genetic Algorithm code

### **Suggested Tools Python/R/MATLAB**

---

<b>Course Code: BTCS620-18</b>	<b>Course Title: Mobile Application Development</b>	<b>L:3; T:0; P:0</b>	<b>3Credits</b>
--------------------------------	---	----------------------	-----------------

### **Details of course:**

#### **Unit-1**

Introduction to Android: The Android Developing environment, Android SDK, Introduction to Open Handset Alliance, Development Framework, Application Fundamentals; Device Compatibility, System permissions, Understanding Anatomy of Android Application, Android Development Tools

**6 hrs., 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

**6 hrs., CO1**

#### **Unit-III**

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

**6 hrs., CO 2**

#### **Unit-IV**

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

**8 hrs., CO 3,4**

#### **Unit-V**

Factors in Developing Mobile Applications: Mobile Software Engineering, Frameworks and Tools, Generic UI Development, Android User

Graphics and Multimedia: Performance and Multithreading, Graphics and UI Performance, Android Graphics, Mobile Agents and Peer-to-Peer Architecture, Android Multimedia

**8 hrs., CO 4,5**

#### **Unit-VI**

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

**8hrs., CO 5**

**Course Outcomes:**

CO 1: Describe those aspects of mobile programming that make it unique from programming for other platforms,

CO 2: Critique mobile applications on their design pros and cons,

CO 3: Utilize rapid prototyping techniques to design and develop sophisticated mobile interfaces,

CO 4: Program mobile applications for the Android operating system that use basic and advanced phone features, and

CO 5: Deploy applications to the Android marketplace for distribution

**References:**

1. Rick Rogers, John Lombardo, Meike Blake, “Android application development”, Ist Edition, O’Reilly, 2010
2. Lauren Darcey and Shane Conder, “Android Wireless Application Development”, 2nd ed. Pearson Education, 2011
3. Wei-Meng Lee , Beginning Android 4 development ,2012 by John Wiley & Sons
4. Jeff Mewherter, Scott Gowell, Wrox Publisher, ”Professional Mobile Application Development”, Ist Edition, 2012
5. Reto Meier, “Professional Android 4 Application Development”, Wrox, 2012

<b>Course Code: BTCS621-18</b>	<b>Course Title: Mobile Application Development Lab</b>	<b>L:0; T:0; P:2</b>	<b>1Credits</b>
--------------------------------	---	----------------------	-----------------

**LIST OF PRACTICALS**

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

## 8. Testing mobile app- unit testing, black box testing and test automation

SEVENTH /  
EIGHTTH  
SEMESTER



<b>Course Code: BTCS 701-18</b>	<b>Course Title : Network Security and Cryptography</b>	<b>3L:0T:0P</b>	<b>3Credits</b>
---------------------------------	---	-----------------	-----------------

### **Detailed Contents:**

#### **UNIT 1: Introduction (3 Hours)**

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

**[5hrs] (CO 1)**

#### **UNIT 2: Math Background**

Modular Arithmetic, Euclidean and Extended Euclidean algorithm, Prime numbers, Fermat and Euler's Theorem

**[5hrs]**

**(CO 1)**

#### **UNIT 3: Cryptography**

Dimensions of Cryptography, Classical Cryptographic Techniques Block Ciphers (DES, AES) : Feistel Cipher Structure, Simplified DES, DES, Double and Triple DES, Block Cipher design Principles, AES, Modes of Operations Public-Key Cryptography : Principles Of Public-Key Cryptography, RSA Algorithm, Key Management, Diffie-Hellman Key Exchange, Elgamal Algorithm, Elliptic Curve Cryptography

**[12hrs] (CO 2)**

#### **UNIT 4 Hash and MAC Algorithms**

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

**[6hrs] (CO 3)**

#### **UNIT 5 Security in Networks**

Threats in networks, Network Security Controls – Architecture, Encryption, Content Integrity, Strong Authentication, Access Controls, Wireless Security, Honeypots, Traffic flow security, Firewalls – Design and Types of Firewalls, Personal Firewalls, IDS, Email Security – PGP, S/MIME

**[7hrs] (CO 4)**

### **Course Outcomes:**

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

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

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

CO3: Apply methods for authentication, access control, intrusion detection and prevention.

CO4: Identify and mitigate software security vulnerabilities in existing systems.

**Suggested Readings/ Books:**

1. Cryptography And Network Security Principles And Practice Fourth Edition, William Stallings, Pearson Education
2. Modern Cryptography: Theory and Practice, by Wenbo Mao, Prentice Hall PTR
3. Network Security Essentials: Applications and Standards, by William Stallings. Prentice Hall
4. Cryptography: Theory and Practice by Douglas R. Stinson, CRC press.

-----  
-----

<b>Course Code: BTCS -702-18</b>	<b>Course Title: Data Warehousing and Data Mining</b>	<b>3L: 0T: 0P</b>	<b>Credits: 3</b>
--------------------------------------	---	-------------------	-------------------

**Detailed Contents:**

**UNIT 1:**

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

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

**[10hrs]**

**UNIT 2:**

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

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

**[10 hrs]**

**UNIT 3:**

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

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

**[10 hrs]**

## **UNIT 4:**

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

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

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

# **ELECTIVE IV**

Course Code: <b>BTCS 704-18</b>	Course Title : <b>Deep Learning</b>	<b>3L:0T:0P</b>	<b>3Credits</b>
---------------------------------	-------------------------------------	-----------------	-----------------

### Detailed Contents:

**UNIT 1: Machine Learning Basics:** Learning, Under-fitting, Overfitting, Estimators, Bias, Variance, Maximum Likelihood Estimation, Bayesian Statistics, Supervised Learning, Unsupervised Learning and Stochastic Gradient Decent.

**[4hrs] (CO 1)**

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

**[4hrs] (CO 2)**

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

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

**[12hrs] (CO 4)**

**UNIT 5: Deep Generative Models:** Boltzmann Machines, Restricted Boltzmann Machines, Deep Belief Networks, Deep Boltzmann Machines, Sigmoid Belief Networks, Directed Generative Net, Drawing Samples from Auto –encoders.

**[14hrs] (CO 5)**

### Course Outcomes:

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

CO1: Comprehend the advancements in learning techniques

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

CO3: Demonstrate the applications of Convolution Networks

CO4: Apply Recurrent Network for Sequence Modelling

CO5: Deploy the Deep Generative Models

### Suggested Readings/ Books:

*Text Books:*

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

*Reference Books:*

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

Course Code: BTCS 705-18	Course Title: Deep Learning Lab	L:0;T:0; 2P:	Credits;1
--------------------------	---------------------------------	-----------------	-----------

**Detailed List of Tasks:**

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

**Suggested Tools Python/R/MATLAB**

---

Course Code: BTCS706-18	Course Title: Distributed Databases	3L: 0T: 0P	Credits: 3
----------------------------	-------------------------------------	------------	------------

**Detailed Contents:**

**Unit 1:**

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

**DISTRIBUTED DATABASE MANAGEMENT SYSTEM ARCHITECTURE:** Transparencies in a distributed DBMS; Distributed DBMS architecture; Global directory issues.

**6 hrs., CO1**

**Unit 2:**

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

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

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

**10 hrs., CO1**

**Unit 3:**

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

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

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

**Unit 4:**

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

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

**ADVANCED TOPICS:** Databases, Distributed Object Management, Multi-databases. **10 hrs., CO2,3**

**COURSE OUTCOMES**

After completion of course, students would be able to:

CO1: Design trends in distributed systems.

CO2: Apply network virtualization in distributed environment.

CO3: Apply remote method invocation and objects.

**References:**

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

<b>Course Code:</b> <b>BTCS707-18</b>	<b>Course Title:</b> Distributed Databases lab	<b>L: T: 2P</b>	<b>Credits: 1</b>
--	---	-----------------	-------------------

## **Detailed list of Tasks:**

Programs may be implemented using any open source tool

**Expt. 1:** Installation and configuration of database packages.

**Expt. 2:** Creating and managing database objects (Tables, views, indexes etc.)

**Expt. 3:** Creating and managing database security through user management.

**Expt. 4:** Creating and maintaining database links.

**Expt. 5:** Implement Partitioning on the database tables.

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

**Expt. 7:** Performance tuning of SQL queries.

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

<b>Course Code:</b> <b>BTCS708-18</b>	<b>Course Title:</b> Computer Vision	<b>3L: 0T: 0P</b>	<b>Credits: 3</b>
--	--------------------------------------	-------------------	-------------------

## **Detailed Contents:**

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

### **Unit 2:**

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

### **Unit 3:**

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

### **Unit 4:**

Feature extraction, shape, histogram, color, spectral, texture, using CVIP tools, Feature analysis, feature vectors, distance /similarity measures, data pre- processing. **8 hrs., CO2**

### **Unit 5:**

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

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

Recent trends in Activity Recognition, computational photography, Biometrics **8 hrs., CO1**

## **COURSE OUTCOMES**

After completion of course, students would be able to:



**CO1:** Understand image detection and analysis

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

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

**References:**

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

<b>Course Code:</b> <b>BTCS708-18</b>	<b>Course Title:</b> Computer Vision lab	<b>L: T: 2P</b>	<b>Credits: 1</b>
--	--	-----------------	-------------------

**Detailed list of Tasks:**

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

**Expt. 1:** Implementation of basic image transformations:

- a. Log
- b. Power law
- c. Negation

**Expt. 2:** Implementation the following:

- a. Histogram processing
- b. Histogram equalization/matching

**Expt. 3:** Implementation of piecewise linear transformations

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

**Expt. 4:** Implementation of image enhancement/smoothing using

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

**Expt. 5:** Implementation of image enhancement/sharpening using

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

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

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

- a. Ideal
- b. Butterworth
- c. Gaussian

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

- a. Ideal
- b. Butterworth
- c. Gaussian

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

- a. Point
- b. Line
- c. Boundary

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

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

<b>Course Code: BTCS 710-18</b>	<b>Course Title : Agile Software Development</b>	<b>3L:0T:0P</b>	<b>3Credits</b>
---------------------------------	--	-----------------	-----------------

### **Detailed Contents:**

#### **UNIT 1: Introduction**

Need of Agile software development, History of Agile, Agile context– manifesto, principles, methods, values. The benefits of agile in software development.

**[6hrs] (CO 1)**

#### **UNIT 2: Agile Design Methodologies**

Fundamentals, Design principles–Single responsibility, Open-closed, Liskov-substitution, Dependency-inversion, Interface-segregation.

**[6hrs] (CO 2)**

#### **UNIT 3: Scrum**

Introduction to scrum framework, Roles: Product owner, team members and scrum master, Events: Sprint, sprint planning, daily scrum, sprint review, and sprint retrospective, Artifacts: Product backlog, sprint backlog and increments. User stories- characteristics and contents.

**[8hrs] (CO 3)**

#### **UNIT 4: Kanban**

Introduction to Kanban framework, Workflow, Limit the amount of work in progress, pulling work from column to column, Kanban board, Adding policies to the board, Cards and their optimization. **[6hrs] (CO 4)**

## **UNIT 5: Extreme Programming**

Basic values and principles, Roles, Twelve practices of XP, Pair programming, XP team, Life cycle and tools for XP.

**[6hrs] (CO 5)**

## **UNIT 6: Agile Testing**

The Agile lifecycle and its impact on testing, Test driven development– Acceptance tests and verifying stories, writing a user acceptance test, Developing effective test suites, Continuous integration, Code refactoring. Risk based testing, Regression tests, Test automation. **[6hrs] (CO 6)**

### **Course Outcomes:**

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

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

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

CO3 Define the core practices behind Scrum framework.

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

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

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

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

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

<b>Course Code:</b> <b>BTCS 711-18</b>	<b>Course Title : Agile Software Development Lab</b>	<b>L:0T:2P</b>	<b>Credits:1</b>
---	--	----------------	------------------

### **Detailed List of Tasks:**

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

# ELECTIVE V

<b>Course Code:</b> <b>BTCS721-18</b>	<b>Course Title: Block Chain Technology</b>	<b>3L:0 T: 0P</b>	<b>Credits: 3</b>
--	---	-------------------	-------------------

## **Detailed Contents:**

### **INTRODUCTION TO BLOCKCHAIN**

Blockchain- Public Ledgers, Blockchain as Public Ledgers -Bitcoin, Blockchain 2.0, Smart Contracts, Block in a Blockchain, Transactions-Distributed Consensus, The Chain and the Longest Chain - Cryptocurrency to Blockchain 2.0 - Permissioned Model of Blockchain, Cryptographic -Hash Function, Properties of a hash function-Hash pointer and Merkle tree

### **BITCOIN AND CRYPTOCURRENCY**

A basic crypto currency, Creation of coins, Payments and double spending, FORTH – the precursor for Bitcoin scripting, Bitcoin Scripts , Bitcoin P2P Network, Transaction in Bitcoin Network, Block Mining, Block propagation and block relay, Consensus introduction, Distributed consensus in open environments- Consensus in a Bitcoin network

### **BITCOIN CONSENSUS**

Bitcoin Consensus, Proof of Work (PoW)- Hashcash PoW , Bitcoin PoW, Attacks on PoW ,monopoly problem- Proof of Stake- Proof of Burn - Proof of Elapsed Time - Bitcoin Miner, Mining Difficulty, Mining Pool-Permissioned model and use cases, Design issues for Permissioned Blockchains, Execute contracts- Consensus models for permissioned blockchain-Distributed consensus in closed environment Paxos

### **DISTRIBUTED CONSENSUS**

RAFT Consensus-Byzantine general problem, Byzantine fault tolerant system-Agreement Protocol, Lamport-Shostak-Pease BFT Algorithm-BFT over Asynchronous systems, Practical Byzantine Fault Tolerance

### **HYPER LEDGER FABRIC & ETHERUM**

Architecture of Hyperledger fabric v1.1-Introduction to hyperledger fabric v1.1, chain code- Ethereum: Ethereum network, EVM, Transaction fee, Mist Browser, Ether, Gas, Solidity, Smart contracts, Truffle Design and issue Crypto currency, Mining, DApps, DAO

### **BLOCKCHAIN APPLICATIONS**

Internet of Things-Medical Record Management System-Block chain in Government and Block chain Security-Block chain Use Cases –Finance

## **COURSE OUTCOMES**

**CO1:** Understand emerging abstract models for Block chain Technology.

**CO2:** Identify major research challenges and technical gaps existing between theory and practice in crypto currency domain.

**CO3:** It provides conceptual understanding of the function of Blockchain as a method of securing distributed ledgers, how consensus on their contents is achieved, and the new applications that they enable.

**CO4:** Apply hyperledger Fabric and Ethereum platform to implement the Block chain Application.

### **REFERENCES**

1. Mastering Blockchain: Deeper insights into decentralization, cryptography, Bitcoin, and popular Blockchain frameworks by Bashir, Imran, 2017.
2. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. Bitcoin and cryptocurrency technologies: a comprehensive introduction. Princeton University Press, 2016.
3. Joseph Bonneau et al, SoK: Research perspectives and challenges for Bitcoin and cryptocurrency, IEEE Symposium on security and Privacy, 2015.

<b>Course Code: 713-18</b>	<b>Course Title: Block chain Technology lab</b>	<b>L: T: 2P</b>	<b>Credits:1</b>
----------------------------	---	-----------------	------------------

1. To Develop Naive Block chain construction.
2. Design Memory Hard algorithm and its Implementation
3. Design Toy application using Blockchain
5. Program to Solve a Mining puzzles using Block chain
6. The ability to formulate mathematical models and problem-solving skills through programming techniques for addressing real-time problems using appropriate data structures and algorithms.
7. The ability to provide design, build, and deploy a distributed application and provide solutions using block chain applications to enhance business measures by sharing information safely and effectively.
8. The ability to create crypto currencies and give a strong technical understanding of Block chain technologies with an in-depth understanding of applications, open research challenges, and future directions.

<b>Course Code: BTCS714-18</b>	<b>Course Title: Parallel Computing</b>	<b>3L: 0T: 0P</b>	<b>Credits: 3</b>
--------------------------------	---	-------------------	-------------------

### **Detailed Contents:**

**Introduction:** Paradigms of parallel computing: Synchronous - vector/array, SIMD, Systolic; Asynchronous - MIMD, reduction paradigm.

**Hardware taxonomy:** Flynn's classifications, Handler's classifications. Software taxonomy: Kung's taxonomy, SPMD.

**Abstract parallel computational models:** Combinational circuits, Sorting network, PRAM models, Interconnection RAMs. Parallelism approaches - data parallelism, control parallelism

**Performance Metrics:** Laws governing performance measurements. Metrics - speedups, efficiency, utilization, communication overheads, single/multiple program performances, bench marks.

**Parallel Processors:** Taxonomy and topology - shared memory mutliprocessors, distributed memory networks. Processor organization - Static and dynamic interconnections. Embeddings and simulations.

**Parallel Programming:** Shared memory programming, distributed memory programming, object oriented programming, data parallel programming, functional and dataflow programming.

**Scheduling and Parallelization:** Scheduling parallel programs. Loop scheduling. Parallelization of sequential programs. Parallel programming support environments.

**Books and References:**

1. M. J. Quinn. Parallel Computing: Theory and Practice , McGraw Hill, New York, 1994.
2. T. G. Lewis and H. El-Rewini. Introduction to Parallel Computing , Prentice Hall, New Jersey, 1992.
3. T. G. Lewis. Parallel Programming: A Machine-Independent Approach , IEEE Computer Society Press, Los Alamitos, 1994.

**Research articles.**

<b>Course Code:</b> <b>BTCS715-18</b>	<b>Course Title: Parallel Computing lab</b>	<b>L: T: 2P</b>	<b>Credits: 1</b>
--	---	-----------------	-------------------

The details may be designed by course instructor as per the theory.

<b>BTCS 716-18</b>	<b>Adhoc and Wireless Sensor Networks</b>	<b>L:03, T:0, P: 0</b>	<b>Credits: 3</b>
--------------------	---	------------------------	-------------------

**Detailed Contents:**

**UNIT 1:**

**{07hrs}(CO1)**

**ADHOC AND SENSORS NETWORKS – INTRODUCTION AND ROUTING PROTOCOLS:**

Wireless Sensor Networks (WSNs): concepts and architectures - Applications of Ad Hoc and Sensor Networks - Design Challenges in Ad hoc and Sensor Networks. Wireless Networks, Issues in Ad hoc wireless networks, Routing Protocol for Ad Hoc Wireless Networks, Classifications of Routing Protocols, Table Driven Routing Protocols – Destination Sequenced Distance Vector (DSDV), On–Demand Routing protocols –Ad hoc On–Demand Distance Vector Routing (AODV).

**UNIT2:**

**{09hrs}(CO2)**

**WSN NETWORKING CONCEPT AND MAC PROTOCOLS :**



Issues in Designing a MAC Protocol for Ad Hoc Wireless Networks - Design Goals of a MAC Protocol for Ad Hoc Wireless Networks, MAC Protocols for wireless sensors Networks, Low duty cycle Protocols and Wakeup concepts, Classification of MAC Protocols , S-MAC, Contention based protocols -PAMAS schedule based protocols –LEACH, IEEE 802.15.4. MAC protocols , Energy efficient routing challenges and issues in transport layer

### **UNIT 3:**

**{06hrs}(CO3)**

#### **ROUTING PROTOCOLS AND TRANSPORT LAYER IN AD HOC WIRELESS NETWORKS:**

Routing Protocol: Issues in designing a routing protocol for Ad hoc networks - Classification- proactive routing - reactive routing (on-demand) - hybrid routing - Transport Layer protocol for Ad hoc networks - Design Goals of a Transport Layer Protocol for Ad Hoc Wireless Networks -Classification of Transport Layer solutions-TCP over Ad hoc wireless ,

### **UNIT4:**

**{06hrs}(CO4)**

#### **SENSOR NETWORKS INTRODUCTION AND ARCHITECTURES:**

Challenges for Wireless Sensor Networks, Enabling Technologies for Wireless Sensor Networks, WSN application examples, Single-Node Architecture – Hardware Components, Energy Consumption of Sensor Nodes, Network Architecture – Sensor Network Scenarios, Transceiver Design Considerations.

### **UNIT 5:**

**{07hrs}(CO5)**

#### **SENSOR NETWORK SECURITY- NETWORK SECURITY :**

Security in Ad Hoc Wireless Networks - Network Security Requirements. Network Security requirements issues and Challenges in security provisioning Network, Security Attacks. Layer wise attack in wireless sensor networks, possible solutions for Jamming, tampering black hole attack, Flooding attack, Key distribution and Management, Secure Routing -SPINS reliability requirements in sensors Networks. Sensor Network Platforms and Tools

### **Course Outcomes:**

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

<b>CO Nos.</b>	<b>Course Outcomes:</b>
<b>CO1</b>	Explain the Fundamental Concepts and applications of ad hoc and wireless sensor networks and apply this knowledge to identify the suitable routing algorithm based on the network.
<b>CO2</b>	Apply the knowledge to identify appropriate physical and MAC layer protocols
<b>CO3:</b>	Understand the transport layer and Describe routing protocols for ad hoc wireless networks with respect to TCP design issues
<b>CO 4</b>	Be familiar with the OS used in Wireless Sensor Networks and build basic modules
	CO 5:

<b>CO 5</b>	Understand the Challenges in security provisioning ,Security Attacks and security issues possible in Adhoc and Sensors Networks
-------------	---

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

##### **Text Books:**

1. C. Siva Ram Murthy, and B. S. Manoj, "Ad Hoc Wireless Networks: Architectures and Protocols ", Pearson Education, 2008.
2. Labiod. H, "Wireless Adhoc and Sensor Networks", Wiley, 2008.
3. 3. Li, X, "Wireless ad -hoc and sensor Networks: theory and applications", Cambridge University Press, 2008.

##### **Reference Books**

1. Carlos De Moraes Cordeiro, Dharma Prakash Agrawal "Ad Hoc & Sensor Networks: Theory and Applications", world Scientific Publishing Company, 2nd edition, 2011.
2. Feng Zhao and Leonides Guibas, "Wireless Sensor Networks", Elsevier Publication
3. Holger Karl and Andreas Willig "Protocols and Architectures for Wireless Sensor Networks", Wiley, 2005 (soft copy available) .
4. Kazem Sohraby, Daniel Minoli, & Taieb Znati, "Wireless Sensor Networks Technology, Protocols, and Applications", John Wiley, 2007. (soft copyavailable).
5. Anna Hac, "Wireless Sensor Network Designs", John Wiley, 2003.(soft copy available)

##### **Online Resources:**

1. [www.wirelessnetworksonline.com](http://www.wirelessnetworksonline.com)
2. [www.securityinwireless.com](http://www.securityinwireless.com)
3. [www.ida.liu.se/~petel71/SN/lecture-notes/sn.pdf](http://www.ida.liu.se/~petel71/SN/lecture-notes/sn.pdf) Practice Aspects 1. NS2 Simulator tool

<b>BTCS 717-18</b>	<b>Adhoc and Wireless Sensor Networks Lab</b>	<b>L:0, T:0, P: 2</b>	<b>Credits: 1</b>
--------------------	---	-----------------------	-------------------

#### List of Experiments :

Sr. No	Name and list of Practical
1	Introduction of Wireless sensor network applications and its simulation
2	Network Simulator installation of wireless sensor network.
3	Implementation of routing protocol in NS2 for DSR protocol
4	Study other wireless sensor network simulators (Mannasim. Contiki
5	Implementation of routing protocol in NS2 for AODV protocol for TORA protocol

<b>Course Code:</b> <b>BTCS718-18</b>	<b>Course Title:</b> Quantum Computing	<b>3L: 0T: 0P</b>	<b>Credits: 3</b>
--	--	-------------------	-------------------

## Detailed Syllabus

**UNIT-1** Fundamental Concepts: Global Perspectives, Quantum Bits, Quantum Computation, Quantum Algorithms, Quantum Information, Postulates of Quantum Mechanisms. 8

Hrs.

**UNIT-II** Quantum Computation: Quantum Circuits – Quantum algorithms, Single Orbit operations, Control Operations, Measurement, Universal Quantum Gates, Simulation of Quantum Systems, Quantum Fourier transform, Phase estimation, Applications, Quantum search algorithms – Quantum counting – Speeding up the solution of NP – complete problems – Quantum Search for an unstructured database.

8 Hrs.

**UNIT-III** Quantum Computers: Guiding Principles, Conditions for Quantum Computation, Harmonic Oscillator Quantum Computer, Optical Photon Quantum Computer – Optical cavity Quantum electrodynamics, Ion traps, Nuclear Magnetic resonance. 8

Hrs.

**UNIT-IV** Quantum Information: Quantum noise and Quantum Operations – Classical Noise and Markov Processes, Quantum Operations, Examples of Quantum noise and Quantum Operations – Applications of Quantum operations, Limitations of the Quantum operations formalism, Distance Measures for Quantum information. 8

Hrs.

**UNIT-V** Quantum Error Correction: Introduction, Shor code, Theory of Quantum Error –Correction, Constructing Quantum Codes, Stabilizer codes, Fault – Tolerant Quantum Computation, Entropy and information – Shannon Entropy, Basic properties of Entropy, Von Neumann, Strong Sub Additivity, Data Compression, Entanglement as a physical resource . 8

Hrs.

### Course Outcomes;

#### At the end of the course students should:

- CO1: understand the quantum model of computation and the basic principles of quantum mechanics;
- CO2: be familiar with basic quantum algorithms and their analysis;
- CO3: be familiar with basic quantum protocols such as teleportation and super dense coding;
- CO4: see how the quantum model relates to classical models of deterministic and probabilistic computation.

### Text books:

1. Micheal A. Nielsen. &Issac L. Chiang, “Quantum Computation and Quantum Information”, Cambridge University Press, Fint South Asian edition, 2002.
2. Eleanor G. Rieffel , Wolfgang H. Polak , “Quantum Computing - A Gentle Introduction” (Scientific and Engineering Computation) Paperback – Import, 3 Oct 2014
3. Computing since Democritus by Scott Aaronson

4. Computer Science: An Introduction by N. David Mermin 5. Yanofsky's and Mannucci, Quantum Computing for Computer Scientists.

<b>Course Code:</b> <b>BTCS719-18</b>	<b>Course Title:</b> Quantum Computing lab	<b>L: T: 2P</b>	<b>Credits: 1</b>
--	---	-----------------	-------------------

1. List modern relevant quantum algorithms and their purposes.
2. Explain the key principles of the various models of quantum computation (circuit, measurement-based, adiabatic model).
3. Explain the basic structure of the quantum algorithms addressed in the course that are based on the circuit model, and to compute the outcome of basic quantum circuits.
4. Compare, in terms of time complexity, what quantum advantage is expected from the quantum algorithms addressed in the course with respect to their classical counterparts.
5. Program simple quantum algorithms on a cloud quantum computer or a cloud simulator.
6. Understand the basic principles of the continuous variable encoding for quantum information processing.
7. Give examples of the motivation for applying quantum computing to machine learning and of what the obstacles are to achieving an advantage from doing so.