# Scheme & Syllabus of

## Bachelor of Technology

## Computer Science & Engg. (BlockChain)/ Comp. Sc. & Design/

## (Batch2023 onwards)

## Bachelor of Technology

## Robotics & Artificial Intelligence

## Batch 2023 onwards

# 3rd & 4th semester


**By**

**Department of Academics**
**( BoS- CSE/ IT)**

**IK Gujral Punjab Technical University**

## Bachelor of Technology in : Computer Science & Engg. (BlockChain)/ Comp. Sc. & Design/ Robotics & Artificial Intelligence

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

**Scheme: Third Semester**

| Course Code | Type of Course | Course Title | Hours per Week | | | Marks Distribution | | Total Marks | Credits |
|---|---|---|---|---|---|---|---|---|---|
| | | | L | T | P | Internal | External | | |
| **BTES 301-18** | Engineering Science Course | Digital Electronics | 3 | 0 | 0 | 40 | 60 | 100 | 3 |
| **BTCS 301-18** | Professional Core Courses | Data structure & Algorithms | 3 | 0 | 0 | 40 | 60 | 100 | 3 |
| **BTCS 302-18** | Professional Core Courses | Object Oriented Programming | 3 | 0 | 0 | 40 | 60 | 100 | 3 |
| **BTAM 302-23** | Basic Science Course | Mathematics-III* (Probability and Statistics) | 4 | 1 | 0 | 40 | 60 | 100 | 3 |
| **HSMC 101/102-18** | Humanities & Social Sciences Including Management \Courses | Foundation Course in Humanities (Development of Societies/Philosophy) | 2 | 1 | 0 | 40 | 60 | 100 | 3 |
| **BTES 302-18** | Engineering Science Course | Digital Electronics Lab | 0 | 0 | 2 | 30 | 20 | 50 | 1 |
| **BTCS 303-18** | Professional Core Courses | Data structure & Algorithms Lab | 0 | 0 | 4 | 30 | 20 | 50 | 2 |
| **BTCS 304-18** | Professional Core Courses | Object Oriented Programming lab. | 0 | 0 | 4 | 30 | 20 | 50 | 2 |
| **BTCS 305-18** | Professional Core Courses | IT Workshop** | 0 | 0 | 2 | 30 | 20 | 50 | 1 |
| | | Summer Institutional Training | 0 | 0 | 0 | 0 | 0 | 0 | Satisfactory/Unsatisfactory |
| **Total** | | | **15** | **2** | **12** | **320** | **380** | **700** | **21** |

* These are the minimum contact hrs. allocated. The contact hrs.may be increased by an institute as per the requirement of the subject.

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

**Fourth Semester**

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

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

# *Third Semester*

| **Course Code:** BTCS301-18 | **Course Title:** Data Structure & Algorithms | **3L:0T:P** | **3Credits** |
|---|---|---|---|

**Detailed Contents:**

**Module 1: Introduction**

Basic Terminologies: Elementary Data Organizations, Data Structure Operations: insertion, deletion, traversal etc.; Analysis of an Algorithm, Asymptotic Notations, Time-Space trade off.

**Searching:** Linear Search and Binary Search Techniques and their complexity analysis.

**[6 hrs] (CO1)**

**Module 2: Stacks and Queues**

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

**[10 hrs] (CO2, CO4, CO5)**

**Module 3**: **Linked Lists**

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

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

**[10 hrs] (CO2, CO4, CO5)**

**Module 4: Sorting and Hashing**

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

**[10 hrs] (CO3)**

**Module 4: Graph**

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

**[6 hrs] (CO2, CO4)**

**Course Outcomes:**

The student will be able to:

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

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

**Suggested Books:**

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

**Reference Books:**

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

---

| **Course Code:** BTCS302-18 | **Course Title:** Object Oriented Programming | **3L:0T:0P** | **3Credits** |
|---|---|---|---|

**Pre-requisites:** Programming in C

**Detailed Contents:**

**Module 1: Introduction**

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

**[8 hrs] (CO1)**

**Module 2: Classes & Objects –II**

Constructors, Destructors, friend functions, Parameterized constructors, Static data members, Functions, Arrays of objects, Pointers to objects, this pointer, and reference parameter, Dynamic allocation of objects, 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**

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

**[10 hrs] (CO5)**

**Course Outcomes:**

The student will be able to:

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

**Suggested Books:**

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

**Reference Books:**

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

---

| **Course Code:** BTCS303-18 | **Course Title:** Data Structure & AlgorithmsLab | **0L:0T:4P** | **2Credits** |
|---|---|---|---|

**List of Experiment:**

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

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

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

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

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

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

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

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

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

**Task 10:** Write a menu driven program to perform following insertion

operations in a single linked list:

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

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

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

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

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

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

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

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

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

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

**Task 19:** Program to traverse graphs using BFS.

**Task 20:** Program to traverse graphs using DFS.

**Lab Outcomes:**

The student will be able to:

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

**Reference Books:**

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

---

| **Course Code:** BTCS304-18 | **Course Title:** Object Oriented Programming Lab | **0L:0T:4P** | **2Credits** |
|---|---|---|---|

**List of Experiment:**

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

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

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

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

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

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

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

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

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

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

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

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

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

**Task 14:** Write a program to demonstrate the multiple inheritances.

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

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

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

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

**Lab Outcomes:**

The student will be able to:

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

**Reference Books:**

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

---

| BTAM302-23 | Mathematics-III (Probability and Statistics) | L-4, T-1, P-0 | 4 Credits |
|---|---|---|---|

**Pre-requisite:** Intermediate Calculus and Basic algebra

**Course Objectives:** The objective of this course is to familiarize the students with fundamental concepts of theory of probability and statistics. The major focus of the course will be on a systematic mathematical treatment of these concepts and their applications.

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

| | |
|---|---|
| **CO1** | Analyze given data using measures of central tendency, skewness and kurtosis. |
| **CO2** | Understand and deal with randomness occurring in real world phenomena. |
| **CO3** | Apply theoretical discrete and continuous probability distributions to deal with real world problems. |
| **CO4** | Analyze given data using the concepts of correlation and regression and fitting of curves. |
| **CO5** | Analyze hypothesis based on small and large samples using different tests of significance. |

**Detailed Content:**

**Unit I**

Measures of Central tendency: Moments, skewness and Kurtosis, Random experiment, Probability axioms, Definition of Probability, conditional probability, Discrete and Continuous random variables, Expectation of Discrete and Continuous random variables.

**Unit II**

Probability distributions: Binomial, Poisson and Normal, Poisson approximation to the binomial distribution, Evaluation of statistical parameters for these three distributions, Bivariate distributions and their properties.

**Unit III**

Correlation and regression for bivariate data, Rank correlation. Curve fitting by the method of least squares, fitting of straight lines, second degree parabolas and more general curves.

**Unit IV**

Test of significances: Sampling and standard error, Tests of significance for large samples and small samples (t-distribution, F-distribution), Chi-square test for goodness of fit and independence of attributes.

**Recommended Books:**

1. S.P. Gupta, Statistical Methods, Sultan Chand & Sons, 33rd Edition, 2005.
2. S.C. Gupta and V. K. Kapoor, Fundamentals of Mathematical Statistics, Sultan Chand & Sons, 2014.
3. S. Ross, A First Course in Probability, 6th Edition, Pearson Education India, 2002.
4. N.P. Bali and Manish Goyal, A text book of Engineering Mathematics, Laxmi Publications, Reprint, 2010.
5. Robert V. Hogg, Joseph W. Mckean and Allen T. Craig, Introduction to Mathematical Statistics, 7th Edition, Pearson, 2012.

**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'sVakyapadiyam)
- Fourteen Knowledge basis as a sources of Vidya: Four Vedas; Six auxiliary sciences (Vedangas); Purana, Nyaya, Mimamsa and Dharma Sastras.

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

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

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

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

### 2.7 Unit 7:
Knowledge about moral and ethics codes.

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

## 3. READINGS

1. Copleston, Frederick, History of Philosophy, Vol. 1.Great Britain: Continuum.
2 Hiriyanna, M. Outlines of Indian Philosophy, MotilalBanarsidass Publishers; Fifth Reprint edition (2009)
3 Sathaye, Avinash, Translation of NasadiyaSukta
4. Ralph T. H. Griffith. The Hymns of the Ṛgveda. MotilalBanarsidass: Delhi: 1973.
5. Raju, P. T. Structural Depths of Indian Thought, Albany: State University of New York Press.
6. Plato, Symposium, Hamilton Press.
7. KautilyaArtha Sastra. Penguin Books, New Delhi.
8. Bacon, Nova Orgum
9. Arnold, Edwin. The Song Celestial.
10. Foucault, Knowledge/Power.
11. Wildon, Anthony, System of Structure.
12. Lele, W.K. The Doctrine of Tantrayukti. Varanasi: Chowkamba Series.
13. Dasgupta, S. N. History of Indian Philosophy, MotilalBanasidas, Delhi.
14. Passmore, John, Hundred Years of Philosophy, Penguin.

**4. OTHER SESSIONS:**

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. Bhattacharys, Daya Krishna, Gopinath Bhattacharya; comparative study of philosophical system such as MadhyasthaDarshan.

**6. OUTCOME OF THE COURSE:**

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

| Course Code:BTES301-18 | Course Title: Digital Electronics | 3L:0T:0P | 3Credits |
|---|---|---|---|

**Detailed Contents:**

**Module 1:**

**NUMBER SYSTEMS:** Binary, Octal, Decimal, Hexadecimal. Number base conversions, 1's, 2's complements, signed Binary numbers. Binary Arithmetic, Binary codes: Weighted BCD, Gray code, Excess 3 code, ASCII.
**LOGIC GATES:** AND, OR, NOT, NAND, NOR, Exclusive-OR and Exclusive-NOR. Implementations of Logic Functions using gates, NAND-NOR implementations.

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

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

**SEQUENTIAL CIRCUITS:** Flip flops SR, JK, T, D and Master slave, Excitation table, Edge triggering, Level Triggering, Realization of one flip flop using other flip flops. Asynchronous/Ripple counters, Synchronous counters, Modulo-n counter, Ring Counters. Design of Synchronous counters: state diagram, Circuit implementation. Shift registers.

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

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

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

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

**Suggested Readings/ Books:**


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


| Course Code:BTES302-18 | Course Title: Digital Electronics Lab | 0L:0T:2P | 1Credits |
|---|---|---|---|

# List of Experiments:

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

**Course Outcomes**

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

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

# *Fourth Semester*

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

**Pre-requisites:** Digital Electronics

**Detailed Contents:**

**Module 1: Functional blocks of a computer**

CPU, memory, input-output subsystems, control unit. Instruction set architecture of a CPU – registers, instruction execution cycle, RTL interpretation of instructions, addressing modes, instruction set. Case study – instruction set of 8085 processor.

**Data representation**: signed number representation, fixed and floating point representations, character representation. Computer arithmetic – integer addition and subtraction, ripple carry adder, carry look-ahead adder, etc. multiplication – shift-andadd, Booth multiplier, carry save multiplier, etc. Division restoring and non-restoring techniques, floating point arithmetic.

**[10 hrs] (CO1, CO2)**

**Module 2: Introduction to x86 architecture.**

**CPU control unit design**: Hardwired and micro-programmed design approaches, Case study – design of a simple hypothetical CPU.

**Memory system design**: semiconductor memory technologies, memory organization.

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

**[12 hrs] (CO2, CO4)**

**Module 3: Pipelining**

Basic concepts of pipelining, throughput and speedup, pipeline hazards.

**Parallel Processors**: Introduction to parallelprocessors, Concurrent access to memory and cache coherency.

**[10 hrs] (CO5)**

**Module 4: Memory Organization**

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

**[10 hrs] (CO3)**

**Course Outcomes:**

The student will be able to:

1. Understand functional block diagram of microprocessor;
2. Apply instruction set for Writingassembly language programs;
3. Design a memory module and analyze its operation by interfacing with the CPU;
4. Classify hardwired and microprogrammed control units; &
5. Understand the concept of pipelining and its performance metrics.

**Suggested Books:**


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

**Reference Books:**


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


---

| **Course Code:** BTCS402-18 | **Course Title:** Operating Systems | **3L:0T:0P** | **3Credits** |
|---|---|---|---|

**Detailed Contents:**

**Module 1: Introduction**

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

**[6 hrs] (CO1)**

**Module 2: Processes**

Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching

**Thread:** Definition, Various states, Benefits of threads, Types of threads, Concept of multithreads,

**Process Scheduling**: Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time; Scheduling algorithms: Pre-emptive and Non-pre-emptive, FCFS, SJF, RR; Multiprocessor scheduling: Real Time scheduling: RM and EDF.

**[10 hrs] (CO2, CO3)**

**Module 3: Inter-process Communication**

Critical Section, Race Conditions, Mutual Exclusion, Hardware Solution, StrictAlternation, Peterson's Solution, TheProducer\ConsumerProblem, Semaphores,EventCounters,Monitors, Message Passing, Classical IPC Problems: Reader's & Writer Problem, Dinning Philosopher Problem etc.

**[8 hrs] (CO2)**

**Module 4: Deadlocks**

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

**[8 hrs] (CO3)**

**Module 5: MemoryManagement**

Basicconcept,LogicalandPhysical address map, Memory allocation: Contiguous Memory allocation –Fixedandvariable partition–Internaland 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, Allocationmethods (contiguous, linked, indexed), Free Space Management (bit vector, linked list, grouping), directory implementation (linear list, hash table), efficiency and performance.

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

**[8 hrs] (CO5, CO6)**

**Course Outcomes:**

The student will be able to:

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

**Suggested Books:**

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

**Reference Books:**

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

| **Course Code:** BTCS403-18 | **Course Title:** Design and Analysis of Algorithms | **3L:0T:0P** | **3Credits** |
|---|---|---|---|

**Pre-requisites:** Data Structures

**Detailed Contents:**

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

**[8 hrs] (CO1)**

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

**[10 hrs] (CO1, CO2)**

**Module 3: Graph and Tree Algorithms**

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

**[10 hrs] (CO3)**

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

**[8 hrs] (CO5)**

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

**[6 hrs] (CO1, CO4, CO5)**

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

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

**Suggested Books:**

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

**Reference Books**


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


---

| **Course Code:** BTES402-18 | **Course Title:** Computer Organization & ArchitectureLab | **0L:0T:2P** | **1Credits** |
|---|---|---|---|

**List of Experiment:**

- **Task 1:** Computer Anatomy- Memory, Ports, Motherboard and add-on cards.
- **Task 2:** Dismantling and assembling PC.
- **Task 3:** Introduction to 8085 kit.
- **Task 4:** 2. Addition of two 8 bit numbers, sum 8 bit.
- **Task 5:** Subtraction of two 8 bit numbers.
- **Task 6:** Find 1's complement of 8-bit number.
- **Task 7:** Find 2's complement of 8-bit number.
- **Task 8:** Shift an 8-bit no. by one bit.
- **Task 9:** Find Largest of two 8 bit numbers.
- **Task 10:** Find Largest among an array of ten numbers (8 bit).
- **Task 11:** Sum of series of 8 bit numbers.
- **Task 12:** Introduction to 8086 kit.
- **Task 13:** Addition and subtraction of two 16 bit numbers, sum 16 bit.
- **Task 14:** Implement of Booth's algorithm for arithmetic operations.
- **Task 15:** Find 1's and 2's complement of 16-bit number.
- **Task 16:** Implement simple programs using I/O based interface.

**Lab Outcomes:**

The student will be able to:

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

**Reference Books:**


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


---

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

**List of Experiment:**

- **Task 1:** Installation Process of various operating systems.
- **Task 2:** Implementation of CPU scheduling algorithms to find turnaround time and waiting time. a) FCFS b) SJF c) Round Robin (pre-emptive) d) Priority.
- **Task 3:** Virtualization, Installation of Virtual Machine Software and installation of Operating System on Virtual Machine.
- **Task 4:** Commands for files & directories: cd, ls, cp, md, rm, mkdir, rmdir. Creating and viewing files using cat. File comparisons. Disk related commands: checking disk free spaces. Processes in linux, connecting processes with pipes, background processing, managing multiple processes. Background process: changing process priority, scheduling of processes at command, batch commands, kill, ps, who, sleep. Printing commands, grep, fgrep, find, sort, cal, banner, touch, file. File related commands ws, sat, cut, grep.
- **Task 5:** Shell Programming: Basic of shell programming, various types of shell, Shell Programming in bash, conditional & looping statement, case statements, parameter passing and arguments, shell variables, shell keywords, creating shell programs for automate system tasks, report printing.
- **Task 6:** Implementation of Bankers algorithm for the purpose of deadlock avoidance.

**Lab Outcomes:**

The student will be able to:

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

**Reference Books:**

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

---

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

**List of Experiment:**

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

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

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

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

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

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

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

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

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

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

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

**Lab Outcomes:**

The student will be able to:

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

**Reference Books**


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


## UNIVERSAL HUMAN VALUES 2: UNDERSTANDING HARMONY

**Course code: HSMC122-18**

Credits: 3

**COURSE TOPICS:**

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

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

1. Purpose and motivation for the course, recapitulation from Universal Human Values-I
2. Self-Exploration–what is it? - Its content and process; 'Natural Acceptance' and ExperientialValidation- as the process 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 ensureSanyam and Health.

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

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

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 appropriatetechnologies 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.

| **Course Code: EVS101-18** | **Course Title: Environmental Studies-** | **L:2; T:0; P:0** | **0Credits** |
|---|---|---|---|

.Detailed Contents

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

Natural resources and associated problems.

a) Forest resources : Use and over-exploitation, deforestation, case studies. Timber extraction, mining, dams and their effects on forest and tribal people.

b) Water resources : Use and over-utilization of surface and ground water, floods, drought, conflicts over water, dams-benefits and problems.

c) Mineral resources : Use and exploitation, environmental effects of extracting and using mineral resources, case studies.

d) Food resources : World food problems, changes caused by agriculture and overgrazing, effects of modern agriculture, fertilizer-pesticide problems, water logging, salinity, case studies.

e) Energy resources : Growing energy needs, renewable and non renewable energy sources, use of alternate energy sources. Case studies.

f) Land resources : Land as a resource, land degradation, man induced landslides, soil erosion and desertification.

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

**Module 2 : Ecosystems**

Concept of an ecosystem. Structure and function of an ecosystem.

Food chains, food webs and ecological pyramids. Introduction, types, characteristic features,

structure and function of following ecosystems:

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

**Module 3 : Biodiversity and its conservation**

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

**Module 4 : Social Issues and the Environment**

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

***ACTIVITIES**

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

Impart knowledge and inculcate the habit of taking interest and understanding biodiversity in and around the college campus. The students should be encouraged to take interest in bird watching, recognizing local plants, herbs and local animals. The students should be encouraged to appreciate the difference in the local biodiversity in their hometown, in the place of their study and other places they visit for vacation/breaks etc.

Following activities must be included.

Identify a tree fruit flower peculiar to a place or having origin from the place.

Making high resolution big photographs of small creatures (bees, spiders, ants. mosquitos etc.) especially part of body so that people can recognize (games on recognizing animals/plants).

Videography/ photography/ information collections on specialties/unique features of different types of common creatures.

Search and explore patents and rights related to animals, trees etc. Studying miracles of mechanisms of different body systems.

**1*(A) Awareness Activities:***

a) Small group meetings about water management, promotion of recycle use, generation of less waste, avoiding electricity waste
b) Slogan making event
c) Poster making event
d) Cycle rally
e) Lectures from experts
***f)*** Plantation
g) Gifting a tree to see its full growth
h) Cleanliness drive
***i)*** Drive for segregation of waste
i) To live with some eminent environmentalist for a week or so to understand his work 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. BharuchaErach, The Biodiversity of India, Mapin Publishing Pvt. Ltd., Ahmedabad – 380 013, India, Email:mapin@icenet.net (R)
3. Brunner R.C., 1989, Hazardous Waste Incineration, McGraw Hill Inc. 480p
4. Clark R.S., Marine Pollution, Clanderson Press Oxford (TB)
5. Cunningham, W.P. Cooper, T.H. Gorhani, E & Hepworth, M.T. 2001, Environmental Encyclopedia, Jaico Publ. House, Mumabai, 1196p
6. Hawkins R.E., Encyclopedia of Indian Natural History, Bombay Natural History Society, Bombay (R)
7. Heywood, V.H &Waston, R.T. 1995. Global Biodiversity Assessment. Cambridge Univ. Press 1140p.
8. Mhaskar A.K., Matter Hazardous, Techno-Science Publication (TB)
9. Miller T.G. Jr. Environmental Science, Wadsworth Publishing Co. (TB)
10. Odum, E.P. 1971. Fundamentals of Ecology. W.B. Saunders Co. USA, 574p
11. Townsend C., Harper J, and Michael Begon, Essentials of Ecology, Blackwell Science (TB)
12. Trivedi R.K., Handbook of Environmental Laws, Rules Guidelines, Compliances and Stadards, Vol I and II, Enviro Media (R)
13. Trivedi R. K. and P.K. Goel, Introduction to air pollution, Techno-Science Publication (TB)
14. Wanger K.D., 1998 Environmental Management. W.B. Saunders Co. Philadelphia, USA 499p


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

**Detailed Contents:**

**Unit I: Social Development** (5 hours)

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

**Unit II: Political Development** (3 hours)

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

**Unit III: Economic Development** (18 hours)

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

Gandhian idea of development. Swaraj and Decentralization.

**PROJECT: Possible projects in this course could be**

a) Interact with local communities and understand their issues.

b) Study local cottage industry and agricultural practices. Role of engineering and specialized knowledge.

c) Evaluation of technology in the context of its application. Social impact of technology. Environmental impact of technology. Evaluation from a holistic perspective.

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

**Detailed Contents:**

**Unit 1:**

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

a. Upanishads;

b. Six systems orthodox and Heterodox Schools of Indian Philosophy.

c. Greek Philosophy:

**Unit 2:**

Origin of the Universe:

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

**Unit 3:**

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

**Unit 4:**

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

**Unit 5:**

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

**Unit 6:**

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

**Unit 7:**

Knowledge about moral and ethics codes.

**Unit 8:**

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

**READINGS**

1. Copleston, Frederick, History of Philosophy, Vol. 1.Great Britain: Continuum.
2 Hiriyanna, M. Outlines of Indian Philosophy, MotilalBanarsidass Publishers; Fifth Reprint edition (2009)
3 Sathaye, Avinash, Translation of NasadiyaSukta
4. Ralph T. H. Griffith. The Hymns of the Ṛgveda. MotilalBanarsidass: Delhi: 1973.
5. Raju, P. T. Structural Depths of Indian Thought, Albany: State University of New York Press.
6. Plato, Symposium, Hamilton Press.
7. KautilyaArtha Sastra. Penguin Books, New Delhi.
8. Bacon, Nova Orgum
9. Arnold, Edwin. The Song Celestial.
10. Foucault, Knowledge/Power.
11. Wildon, Anthony, System of Structure.
12. Lele, W.K. The Doctrine of Tantrayukti. Varanasi: Chowkamba Series.
13. Dasgupta, S. N. History of Indian Philosophy, MotilalBanasidas, Delhi.

14. Passmore, John, Hundred Years of Philosophy, Penguin.

**ASSESSMENT (indicative only):**

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

**OUTCOME OF THE COURSE:**

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

| BTCS401-18 | Discrete Mathematics | 3L:1T:0P | 4 Credits |
|---|---|---|---|

**Detailed contents:**

**Module 1:**

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

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

**Module 2:**

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

**Module 3:**

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

**Module 4:**

**Algebraic Structures and Morphism:** Algebraic Structures with one Binary Operation, Semi Groups, Monoids, Groups, Congruence Relation and Quotient Structures, Free and Cyclic Monoids and Groups, Permutation Groups, Substructures, Normal Subgroups, Algebraic Structures with two Binary Operation, Rings, Integral Domain and Fields. Boolean Algebra and Boolean Ring, 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.

---

# B.Tech Computer Science and Engineering (Block Chain)

## 5th Semester

| Course Code | Type of Course | Course Title | Hours per Week | | | Marks Distribution | | Total Marks | Credits |
|---|---|---|---|---|---|---|---|---|---|
| | | | L | T | P | Internal | External | | |
| BTCS 501-18 | Professional Core Courses | Database Management Systems | 3 | 0 | 0 | 40 | 60 | 100 | 3 |
| BTCS 502-18 | Professional Core Courses | Formal Language & Automata Theory | 3 | 0 | 0 | 40 | 60 | 100 | 3 |
| BTCS 504-18 | Professional Core Courses | Computer Networks | 3 | 0 | 0 | 40 | 60 | 100 | 3 |
| BTBC 501-23 | Professional Core Courses | Fundamentals of Blockchain Technologies | 3 | 0 | 0 | 40 | 60 | 100 | 3 |
| BTCS 503-18 | Professional Core Courses | Software Engineering | 3 | 0 | 0 | 40 | 60 | 100 | 3 |
| **BTCS XXX-XX** | Professional Elective | Elective-I | 3 | 0 | 0 | 40 | 60 | 100 | 3 |
| **MC** | Mandatory Courses | Constitution of India/ Essence of Indian Traditional Knowledge | 2 | - | - | 100 | - | 100 | S/US |
| BTCS 505-18 | Professional Core Courses | Database Management Systems Lab | 0 | 0 | 4 | 30 | 20 | 50 | 2 |
| BTCS 507-18 | Professional Core Courses | Computer Networks Lab | 0 | 0 | 2 | 30 | 20 | 50 | 1 |
| BTCS 506-18 | Professional Core Courses | Software Engineering Lab | 0 | 0 | 2 | 30 | 20 | 50 | 1 |
| BTBC 502-23 | Professional Core Courses | Fundamentals of Blockchain Technologies Lab | 0 | 0 | 2 | 30 | 20 | 50 | 1 |
| **BTCS XXX-XX** | Professional Elective | Elective-I Lab | 0 | 0 | 2 | 30 | 20 | 50 | 1 |
| **Total** | | | 20 | 0 | 12 | 490 | 460 | 950 | 24 |

**Elective-I:**

BTCS 510-18 Programming in Python

BTCS 513-18 Programming in Python Lab

BTIT 503-18 Programming in Java

BTIT 506-18 Programming in Java Lab

BTBC 503-23 Cyber Security

BTBC 504-23 Cyber Security Lab

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

Detailed Contents:

Module 1: Database system architecture

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

[7hrs] (CO1,2)

Module 2: Relational query languages

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

[10hrs] (CO2,4)

Module 3:

Storage strategies, Indices, B-trees, hashing.

[3hrs] (CO3)

Module 4: Transaction processing

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

[6hrs] (CO3)

Module 5: Database Security

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

[8hrs] (CO 4,5)

Module 6: Advanced Topics

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

[8hrs] (CO 5)

Course Outcomes:

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

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

Text Books:

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

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

Sudarshan, McGraw-Hill.

Reference Books:

1. “Principles of Database and Knowledge–Base Systems”, Vol1 by J. D. Ullman, Computer Science Press.

2. “Fundamentals of Database Systems”, 5th Edition by R. Elmasri and S. Navathe, Pearson Education.

3. “Foundations of Databases”, Reprint by Serge Abiteboul, Richard Hull, Victor Vianu, Addison-Wesley.

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

Detailed Contents

Module 1: Introduction

Alphabet, languages and grammars, productions and derivation, Chomsky hierarchy of languages.

[3hrs] (CO1 )

Module 2: Regular languages and finite automata:

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

[8hrs] (CO2 )

Module 3: Context-free languages and pushdown automata

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

[8hrs] (CO3 )

Module 4: Context-sensitive languages

Context-sensitive grammars (CSG) and languages, linear bounded automata and equivalence with CSG.

[5hrs] (CO4 )

Module 5: Turing machines

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

[8hrs] (CO 5 )

Module 6: Undecidability & Intractablity:

Church-Turing thesis, universal Turing machine, the universal and diagonalization languages, reduction between languages and Rice s theorem, undecidable problems about languages. Intractablity: Notion of tractability/feasibility. The classes NP and co-NP, their importance. Polynomial time many-one reduction. Completeness under this reduction. Cook-Levin theorem: NP-completeness of propositional satisfiability, other variants of satisfiability. NP-

complete problems from other domains: graphs (clique, vertex cover, independent sets, Hamiltonian cycle), number problem (partition), set cover

[12hrs] (CO5 )

Course Outcomes: The student will be able to:

CO1: Write a formal notation for strings, languages and machines.
CO2: Design finite automata to accept a set of strings of a language.
CO3: Design context free grammars to generate strings of context free language .
CO4: Determine equivalence of languages accepted by Push Down Automata and languages generated by context free grammars
CO5: Distinguish between computability and non-computability and Decidability and undecidability.

Text Books:

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

Reference Books:

1. Harry R. Lewis and Christos H. Papadimitriou, Elements of the Theory of Computation, Pearson Education Asia.

2. Dexter C. Kozen, Automata and Computability, Undergraduate Texts in Computer Science, Springer.

3. Michael Sipser, Introduction to the Theory of Computation, PWS Publishing.

4. John Martin, Introduction to Languages and The Theory of Computation, Tata McGraw Hill.

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

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.

# Fundamentals of Blockchain Technologies

Course Code: BTBC 501-23 3L:0T, 3 Credits

## Unit 1: 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 Etherum 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.

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

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] (CO2 )

Module 3:

Fundamentals of testing, White-box, and black-box testing, Test coverage analysis and test case design techniques, mutation testing, Static and dynamic analysis, Software reliability metrics, reliability growth modeling. [10hrs] (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”, Adison Wesley, 1996.

2. Watts Humphrey, ”Managing software process”, Pearson education, 2003.

3. James F. Peters and Witold Pedrycz, “ Software Engineering – An Engineering Approach”, Wiley.

4. Mouratidis and Giorgini. “Integrating Security and Software Engineering–Advances and Future”, IGP. ISBN – 1-59904-148-0.

5. Pankaj Jalote, “An integrated approach to Software Engineering”, Springer/Narosa.

6. Fundamentals of Software Engineering by Rajib Mall, – PHI-3rd Edition, 2009.

Course Outcomes:

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

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

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

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

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

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

---

Course Code: BTCS505-18 Course Title: Database management System lab 0L:0T:4P 2Credits

List of Experiments:

Task 1: Introduction to SQL and installation of SQL Server / Oracle.
Task 2: Data Types, Creating Tables, Retrieval of Rows using Select Statement, Conditional Retrieval of Rows, Alter and Drop Statements.
Task 3: Working with Null Values, Matching a Pattern from a Table, Ordering the Result of a Query, Aggregate Functions, Grouping the Result of a Query, Update and Delete Statements.
Task 4: Set Operators, Nested Queries, Joins, Sequences.
Task 5: Views, Indexes, Database Security and Privileges: Grant and Revoke Commands, Commit and Rollback Commands.
Task 6: PL/SQL Architecture, Assignments and Expressions, Writing PL/SQL Code, Referencing Non-SQL parameters.
Task 7: Stored Procedures and Exception Handling.
Task 8: Triggers and Cursor Management in PL/SQL. Suggested Tools – MySQL, DB2, Oracle, SQL Server 2012, Postgre SQL, SQL lite

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

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

---

List of Experiments:

Task 1: To study the different types of Network cables and network topologies.
Task 2: Practically implement and test the cross-wired cable and straight through cable using
clamping tool and network lab cable tester.
Task 3: Study and familiarization with various network devices.
Task 4: Familiarization with Packet Tracer Simulation tool/any other related tool.
Task 5: Study and Implementation of IP Addressing Schemes
Task 6: Creation of Simple Networking topologies using hubs and switches
Task 7: Simulation of web traffic in Packet Tracer
Task 8: Study and implementation of various router configuration commands
Task 9: Creation of Networks using routers.
Task 10: Configuring networks using the concept of subnetting
Task 11: Practical implementation of basic network command and Network configuration commands like ping, ipconfig, netstat, tracert etc. for troubleshooting network related problems.
Task 12: Configuration of networks using static and default routes.

Course Outcomes:

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

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

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

List of Experiments:

Task 1: Study and usage of OpenProj or similar software to draft a project plan

Task 2: Study and usage of OpenProj or similar software to track the progress of a project

Task 3: Preparation of Software Requirement Specification Document, Design Documents and Testing Phase

Task 4: related documents for some problems

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

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

Task 7: To perform unit testing and integration testing

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

Task 9: Testing of a web site

Suggested Tools - Visual Paradigm, Rational Software Architect. Visio, Argo UML, Rational

Application Developer etc. platforms.

---

| Course Code: BTBC502-23 | Course Title: Fundamentals of Blockchain Technologies Lab lab | 0L:0T:2P | 1Credits |
|---|---|---|---|

Lab work using business intelligence tools viz. advanced ms-excel, BI- tools etc. based on theorytopics will be covered by the instructor

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

Detailed Contents:

Module 1:

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

[8hrs] (CO1)

Module 2:

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

Exceptions: Exceptions in Python, Detecting and Handling Exceptions, Context Management,

*Exceptions as Strings, Raising Exceptions, Assertions, Standard Exceptions, *Creating

Exceptions, Why Exceptions (Now)?, Why Exceptions at All?, Exceptions and the sys Module,

Related Modules

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

[10hrs] (CO1,2)

Module 3:

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

[8hrs] (CO 2,3)

Module 4:

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

Related Modules and Other GUIs

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

[10hrs] (CO 4,6)

Module 5:

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

Text Books:

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

Course Outcomes:

The students should be able to:

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

CO2: Demonstrate proficiency in handling Strings and File Systems.

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

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

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

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

Prerequisites: Students should install Python.

List of Experiments:

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

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

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

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

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

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

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

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

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

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

loop *

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

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

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

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

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

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

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

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

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

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

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

# Programming in Java

Course Code: BTIT 503-18 3L:0T, 3 Credits

Unit 1:

Overview: Object oriented programming principles, Java essentials, java virtual machine, program structure in java ,Java class libraries, Data types, Variables and Arrays, Data types and casting, automatic type promotion in expressions, arrays.

Operators and Control Statements: Arithmetic operators, bit wise operators, relational operators, Boolean logical operators, the ? Operator, operator precedence, Java's selection statements, iteration statements, jump statements.

UNIT 2:

Introduction to Classes: Class fundamentals, declaring class, creating objects, introducing methods: method declaration, overloading, using objects as parameters, recursion, Constructors, this keyword, garbage collection, the finalization. [9hrs., CO1]

UNIT 3:

Inheritance: Inheritance basics, using super and final, method overriding, dynamic method dispatch, Abstract Class, Interface: variables and extending Interfaces, Package: Creating and importing packages, Package access protection, Exception Handling: Exception handling fundamentals, Exception types, Uncaught Exceptions Using try and catch, multiple catch clauses, nested try statements, throw, Java’s built-in exceptions. [12hrs.,CO1,2]

UNIT 4:

Multithreaded Programming: The Java thread model, the main thread, creating thread, creating multiple threads, using isAlive () and join (), Thread priorities, synchronization, Inter thread communications, suspending resuming and stopping threads. [4hrs., CO3]

UNIT5:

I/O : I/O Basics, Reading Console Input, Writing Console Output, Reading and Writing Files, Applets: Applet Fundamentals, Applet Architecture, The HTML Applet tag, Passing parameters to Applets., Networking: Networking basics, Java and the Net, TCP/IP Client Sockets URL, URL Connection, TCP/IP Server Sockets, Database connectivity. [6hrs., CO4]

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

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

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

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

CO4. Get exposure to advance concepts like socket and databease connectivity.

Suggested Readings/Books :

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

2.Joyce Farrell, Java for Beginners, Cengage Learning.

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

4.James Edward Keogh, Jim Keogh, J2EE: The complete Reference, McGrawHill

5.Khalid A. Mughal, Torill Hamre, Rolf W. Rasmussen, Java Actually, Cengage Learning.

6. Shirish Chavan, Java for Beginners, 2nd Edition, Shroff Publishers

Programming in Java Lab

Course Code: BTIT 506-18 0L:0T,2P 1 Credits

1. WAP in Java to show implementation of classes.

2. WAP in Java to show implementation of inheritance.

3. WAP in Java to show Implementation of packages and interfaces.

To accomplish CO2;

4. WAP in Java to show Implementation of threads.

5. WAP in Java Using exception handling mechanisms.

6. WAP in Java to show Implementation of Applets.

To accomplish CO3;

7. WAP in Java to show Implementation of mouse events, and keyboard events.

8. WAP in Java to show Implementing basic file reading and writing methods.

9. Using basic networking features,WAP in Java

To accomplish CO4;

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

Cyber Security

Course Code: BTBC 503-23 3L:0T, 3 Credits

Module 1: Introduction to Cyber Security

Computer Security, Threats, Harm, Vulnerabilities, Controls, Authentication, Access Control and Cryptography. Web attack: Browser Attacks, Web Attacks Targeting Users, Obtaining User or Website Data, Email Attacks.

Network Vulnerabilities: Overview of vulnerability scanning, Open Port/Service Identification, Banner/Version Check, Traffic Probe, Vulnerability Probe, Vulnerability Examples, OpenVAS, Metasploit. Network Sniffers and Injection tools.

Module 2: Network Defence tools.

Firewalls and Packet Filters: Firewall Basics, Packet Filter Vs Firewall, Packet Characteristic to Filter, Stateless Vs Stateful Firewalls, Network Address Translation (NAT) and Port Forwarding. VPN: the basic of Virtual Private Networks.

Module 3: Web Application Tools

Scanning for web vulnerabilities tools: Nikto, W3af HTTP utilities - Curl, OpenSSL and Stunnel. Application Inspection tools Zed Attack Proxy, Sqlmap, DVWA, Webgoat. Password Cracking and Brute-Force Tools: John the Ripper, Pwdump, HTC-Hydra.

Module 4: Cyber Attacks, law and Investigations

Hacking, Attack vectors, Cyberspace and Criminal Behaviour, Digital Forensics, Computer Language, Network Language, Realms of the Cyber world. Firewalls and Packet Filters, password Cracking, Keyloggers and Spyware, Virus and Warms, Trojan and backdoors, Steganography, DOS and DDOS attack, SQL injection, Buffer Overflow, Attack on wireless Networks.

Reference Books:


- Edward Amoroso, "Cyber Attacks: Protecting National Infrastructure", Elsevier, 2012.

Margaret J. Goldstein – Martin Gitlin "Cyber Attacks" Twenty First Century Books 2015

| Course Code: BTBC 504-23 | Course Title: Cyber Security Lab | 0L:0T:2P | 1Credits |
|---|---|---|---|

Lab work using business intelligence tools viz. advanced ms-excel, BI- tools etc. based on theorytopics will be covered by the instructor.

# B.Tech Computer Science and Engineering (Block Chain)

## 6th Semester

| Course Code | Type of Course | Course Title | Hours per Week | | | Marks Distribution | | Total Marks | Credits |
|---|---|---|---|---|---|---|---|---|---|
| | | | L | T | P | Internal | External | | |
| BTCS 601-18 | Professional Core Courses | Compiler Design | 3 | 0 | 0 | 40 | 60 | 100 | 3 |
| BTCS 602-18 | Professional Core Courses | Artificial Intelligence | 3 | 0 | 0 | 40 | 60 | 100 | 3 |
| XXXX XXX-XX | Professional Elective Courses | Elective-II | 3 | 0 | 0 | 40 | 60 | 100 | 3 |
| XXXX XXX-XX | Professional Elective Courses | Elective-III | 3 | 0 | 0 | 40 | 60 | 100 | 3 |
| BTOE XXX-XX | Open Elective Courses | Open Elective-I | 3 | 0 | 0 | 40 | 60 | 100 | 3 |
| BTCS XXX-XX | Project | Project-1 | 0 | 0 | 6 | 60 | 40 | 100 | 3 |
| BTCS 604-18 | Professional Core Courses | Compiler Design Lab | 0 | 0 | 2 | 30 | 20 | 50 | 1 |
| BTCS 605-18 | Professional Core Courses | Artificial Intelligence Lab | 0 | 0 | 2 | 30 | 20 | 50 | 1 |
| XXXX XXX-XX | Professional Core Courses | Elective-II Lab | 0 | 0 | 2 | 30 | 20 | 50 | 1 |
| XXXX XXX-XX | Professional Core Courses | Elective-III Lab | 0 | 0 | 2 | 30 | 20 | 50 | 1 |
| **Total** | | | 20 | 0 | 12 | 380 | 420 | 800 | 22 |

**Elective-II**

BTCS 612-18 Cloud computing

BTCS 613-18 Cloud computing lab

BTBC 601-23 Bitcoin and Cryptocurrency

BTBC 602-23 Bitcoin and Cryptocurrency Lab

BTBC 603-23 Blockchain Architecture Design

BTBC 604-23 Blockchain Architecture Design Lab

**Elective-III**

BTBC 605-23 SMART CONTRACTS USING ETHEREUM

BTBC 606-23 SMART CONTRACTS USING ETHEREUM Lab

BTBC 607-23 Implementing Blockchain on Cloud

BTBC 608-23 Implementing Blockchain on Cloud Lab

BTBC 609-23 Cyber Security in Blockchain Technology

BTBC 610-23 Cyber Security in Blockchain Technology Lab

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

Detailed Contents:

UNIT 1: Unit I Introduction to Compilers:

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

Unit II :Syntax Analysis:

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

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

Unit IV: Run-Time Environment and Code Generation:

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

Unit V: Code Optimization:

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

Course Outcomes:

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

CO1: Build concepts on lexical analysis.

CO2: Understand strategies of syntax analysis.

CO3: Learn techniques of Intermediate code generation.

CO4: Undestand code design issues and design code generator.

CO5: Design and develop optimized codes.

Suggested Readings/ Books:


1. A.V. Aho, Monica, R.Sethi, J.D.Ullman, “Compilers, Principles, Techniques and Tools”, Second Edition, Pearson Education/Addison Wesley, 2009.
2. 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.

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

Detailed Contents:

UNIT 1: Introduction (3 Hours)

Concept of AI, history, current status, scope, agents, environments, Problem Formulations, Review of tree and graph structures, State space representation, Search graph and Search tree. [8hrs] (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 uncertainity

CO4: Learning optimization and inference algorithms for model learning

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

Suggested Readings/ Books:

1.Stuart Russell and Peter Norvig, “Artificial Intelligence: A Modern Approach” , 3rd Edition, Prentice Hall

2. Elaine Rich and Kevin Knight, “Artificial Intelligence”, Tata McGraw Hill

3. Trivedi, M.C., “A Classical Approach to Artifical Intelligence”, Khanna Publishing House, Delhi.

4. Saroj Kaushik, “Artificial Intelligence”, Cengage Learning India,

5. David Poole and Alan Mackworth, “Artificial Intelligence: Foundations for Computational Agents”, Cambridge University Press 2010

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

List of Experiments

1 Design a lexical analyser for given language and the lexical analyser should ignore redundant spaces, tabs and new lines. It should also ignore comments. Although the syntax specification states that identifiers can be arbitrarily long, you may restrict the length to some reasonable value. Simulate the same in C language.

2 Write a C program to identify whether a given line is a comment or not.

3 Write a C program to recognize strings under 'a', 'a*b+', 'abb'.

4 Write a C program to test whether a given identifier is valid or not.

5 Write a C program to simulate lexical analyzer for validating operators.

6 Implement the lexical analyzer using JLex, flex or other lexical analyzer generating tools.

7 Write a C program for implementing the functionalities of predictive parser for the mini language specified in Note 1.

8 a) Write a C program for constructing of LL (1) parsing. b) Write a C program for constructing recursive descent parsing.

9 Write a C program to implement LALR parsing.

10 a) Write a C program to implement operator precedence parsing. b) Write a C program to implement Program semantic rules to calculate the expression that takes an expression with digits, + and * and computes the value.

11 Convert the BNF rules into YACC form and write code to generate abstract syntax tree for the mini language specified in Note 1.

12 Write a C program to generate machine code from abstract syntax tree generated by the parser. The instruction set specified in Note 2 may be considered as the target code.

**Course Code: BTCS 605-18 Course Title Artificial Intelligence Lab L:0;T:0;2 P: 1 Credits**

Detailed List of Tasks:

1.Write a programme to conduct uninformed and informed search.

2. Write a programme to conduct game search.

3. Write a programme to construct a Bayesian network from given data.

4. Write a programme to infer from the Bayesian network.

5. Write a programme to run value and policy iteration in a grid world.

6. Write a programme to do reinforcement learning in a grid world

**Course Code: BTCS 612-18 Course Title: Cloud Computing 3L:0T:0P 3Credits**

Detailed Contents:

UNIT1: Introduction :

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

UNIT2: Cloud computing concepts:

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

UNIT 3: Cloud service models:

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

UNIT 4: Cloud deployment models:

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

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

Course Outcomes:

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

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

Suggested Readings/ Books:

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

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

**Course Code: BTCS 612-18 Course Title: Cloud Computing Lab L:0;T:0; P:2 1 Credits**

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

**BTBC 601-23 BITCOIN AND CRYPTOCURRENCIES 3 L 0 T 0 P, 3 C**

Unit 1: INTRODUCTION TO CRYPTO AND CRYPTOCURRENCIES

Introduction, Cryptographic Hash Functions, Hash Pointers and Data Structures, Digital Signatures, Public Keys as Identities, A Simple Cryptocurrency

Unit 2: BITCOIN BASICS

Bitcoin Protocol and Consensus: A High Level Overview, Bitcoin and Blockchain History, Bitcoin Mechanics and Optimizations: A Technical Overview, Bitcoin IRL: Wallets, Mining, and More

Unit 3: MECHANICS OF BITCOIN

Bitcoin Transactions, Bitcoin Scripts, Applications of Bitcoin Scripts, Bitcoin Blocks, The Bitcoin Network, Limitations & Improvements

Unit 4: STORE AND USE BITCOINS

How to Store and Use Bitcoins, Hot and Cold Storage, Online Wallets and Exchanges, Payment Services, Transaction Fees, Currency Exchange Markets

Unit 5: APPLICATIONS AND SCALING

Enabling a Decentralized Future, Distributed Systems and Alternative Consensus, How to Destroy Bitcoin, Crypto economics and Proof-of- State, Scaling Blockchain: Cryptocurrencies for the Masses, Enterprise Blockchain: Real-World Applications, Anonymity: Mixing and Altcoins, Conclusion: Future of Blockchains

Text books


● Antonopoulos, Mastering Bitcoin: Unlocking Digital Cryptocurrencies

● Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System


Other References


● Wattenhofer, The Science of the Blockchain

● Dr. Gavin Wood, “ETHEREUM: A Secure Decentralized Transaction Ledger, ”Yellow paper.2014.

**BTBC 602-23 BITCOIN AND CRYPTOCURRENCIES LAB 0L 0T 2P 1C**

Experiment based upon the theory subject.

**BTBC 603-23 Blockchain Architecture Design 3 L 0 T 0 P, 3 C**

Unit 1: Introduction to Blockchain: Digital Money to Distributed Ledgers , Design Primitives: Protocols, Security, Consensus, Permissions, Privacy. Blockchain Architecture and Design: Basic crypto primitives: Hash, Signature,) Hashchain to Blockchain, Basic consensus mechanisms

Unit II: Consensus: Requirements for the consensus protocols, Proof of Work (PoW), Scalability aspects of Blockchain consensus protocols, Permissioned Blockchains: Design goals, Consensus protocols for Permissioned Blockchains

Unit III: Hyperledger Fabric (A): Decomposing the consensus process , Hyperledger fabric components, Chaincode Design and Implementation

Hyperledger Fabric (B): Beyond Chaincode: fabric SDK and Front End

Unit IV: Use case 1 : Blockchain in Financial Software and Systems (FSS): (i) Settlements, (ii) KYC, (iii) Capital markets, (iv) Insurance

Use case 2: Blockchain in trade/supply chain: (i) Provenance of goods, visibility, trade/supply chain finance, invoice management discounting, etc

Unit V: Use case 3: Blockchain for Government: (i) Digital identity, land records and other kinds of record keeping between government entities, (ii) public distribution system social welfare systems, Blockchain Cryptography, Privacy and Security on Blockchain

Text books:

1. Mastering Bitcoin: Unlocking Digital Cryptocurrencies, by Andreas Antonopoulos

2. Blockchain by Melanie Swa, O'Reilly

3. Hyperledger Fabric - https://www.hyperledger.org/projects/fabric

4. Zero to Blockchain - An IBM Redbooks course, by Bob Dill, David Smits - https://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/crse0401.html

**BTBC 604-23 Blockchain Architecture Design LAB 0L 0T 2P 1C**

Experiment based upon the theory subject.

**BTBC 605-23 SMART CONTRACTS USING ETHEREUM 3 L 0 T 0 P, 3 C**

Unit 1: INTRODUCTION TO SMART CONTRACTS
Smart Contract Basics: Why Smart Contracts? Contract lifecycle, Solidity: Structure, Basic Data Types & Statements, Contract lifecycle, distinction between a payment system and a decentralized applications platform

Unit 2: ETHEREUM
Ethereum – Introduction, Multitude of clients in Ethereum, Production and test networks in Ethereum, Public, private and development deployments, Comparing Bitcoin and Ethereum, Ethereum sub-protocols

Unit 3: SOLIDITY
Demonstration of smart contract, Introduction to Solidity, Solidity in depth, Building blocks, Contract lifecycle, Solidity for Contract Writing, Developing, Compiling and Deploying MyContract, Interacting with the Contract, Limitations of Remix

Unit 4: DECENTRALIZATION
Decentralized Autonomous Organization (DAO), Decentralized Applications, A Central Bank or Your Own Coin, A Crowdfunding System, State, Merkle Patricia Tree, Client Applications , Objects of smart contracts

Unit 5: USE AND APPLICATION OF SMART CONTRACTS
Examples of using smart contracts, Time Elements in developing smart contracts, Features of smart contracts: Autonomy, Trust, Savings, Safety, Efficiency, Other smart contract platforms, Quality of decentralized applications, Code patterns, Discussion of future prospects

Text books

● Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction A. Narayanan, J. Bonneau, E. Felten, A. Miller, S. Goldfeder Princeton University Press 2016

● Mastering Bitcoin by Andreas Antonopoulos, O'Reilly Publishing 2014

Other References

● Ethereum White Paper Vitalik Buterin Online 2017

● Ethereum documentation (http://www.ethdocs.org/en/latest)

● Solidity documentation ((https://solidity.readthedocs.io/en/develop))

**BTBC 606-23 SMART CONTRACTS USING ETHEREUM LAB 0L 0T 2P 1C**

Experiment based upon the theory subject.

**BTBC 607-23 IMPLEMENTING BLOCK CHAIN ON CLOUD 3 L 0 T 0 P, 3 C**

Unit 1: INTRODUCTION

Why Blockchain?, IBM blockchain platform introduction, benefits and differentiators of deploying and using a blockchain environment of LinuxONE, Kubernetes(K8s), IBM cloud private, Gluster FS, IBM secure service container, IBM blockchain platform, Secure service container partition, IBM cloud private cluster.

Unit 2: PLANNING FOR INSTALLATION

Why secure service container? Persistant storage provider, setting up file storage system

IBM blockchain platform console, Minimum network , Pilot network, Production network, Component containers, Resource reallocation, Consideration for specific use cases

Unit 3: SECURE SERVICE CONTAINER INSTALLATION AND CONFIGURATION

Secure service container architecture, SSC bootleader overview, download the image, Hardware requirement for SSC partition, Networking, Supported operating system and platform, software requirement, supported docker version, Supported IBM Cloud Private Versions, required ports, Creating SSC partitions, Installing IBM cloud private cluster, Deploying IBM cloud private, Uninstalling ICP and SSC, Updating cluster resource dynamically

Unit 4: IBM BLOCKCHAIN PLATFORM INSTALLATIONS AND CONFIGURATIONS

Loading Helm chart, setting up role based access control (RBAC) rules, scripted console installation, manual console installation, Creating peer organization, creating a peer, creating the ordering service, Open shift support, Troubleshooting the installation

Unit 5: PERFORMANCE AND CONSIDERATIONS

Application client, Smart contract programming language, Endorsement policy, Orderer block configuration, Peer container resource allocation, Hiper sockets, Hiper socket benefits, Cryptography importance in block chain, CPACF's role in acceleration and protection

Text books

● Serious Cryptography: A Practical Introduction to Modern Encryption By Jean-Philippe Aumasson

● Handbook of Research on Blockchain Technology by Saravanan Krishnan, Valentina Emilia Balas, Julie Golden, Y. Harold Robinson, S. Balaji, Raghvendra Kumar

**BTBC 608-23 IMPLEMENTING BLOCK CHAIN ON CLOUD LAB 0L 0T 2P 1C**

Experiment based upon the theory subject.

**BTBC 609-23 Cyber Security in Blockchain Technology 3 L 0 T 0 P, 3 C**

Unit 1: Privacy, Security issues in Blockchain

Pseudo-anonymity vs. anonymity, Zcash and Zk-SNARKS for anonymity preservation attacks on Blockchains – such as Sybil attacks, selfish mining, 51% attacks - -advent of algorand Sharding based consensus algorithms to prevent these attacks

Unit 2: Cryptography

Public Key Infrastructure (PKI) and Cryptography, Conventional PKI , Blockchain as a Form of Distributed PKI , Blockchain vs PKI, Blockchain - Public Key Cryptography, Decentralized Public Key Infrastructure (DPKI)

Unit 3: Digital Signature

Digital Signature from Blockchain context, Undeniable signature, Diffie–Hellman, Digital signature scheme for information non-repudiation in blockchain

Unit 4: Blockchain-based time stamping

Time stamping Metadata Using Blockchain, Decentralized Trusted Time stamping Based on Blockchains, Content Time stamping

Unit 5: Use Cases of Blockchain In Cyber security

Decentralized Storage Solutions, How Guardtime uses blockchain technology to safeguard data, IoT Security, Safer DNS, Using blockchains to prevent DDoS attacks, Implementing Security in Private Messaging

**Text book:**

Blockchain Technology Basics: Blockchain cryptography and cybersecurity Kindle Edition by Raghava Shankar, Srikanth RC Cherukupalli

**Other References**

Mastering Blockchain: Deeper insights into decentralization, cryptography, Bitcoin, and popular Blockchain frameworks Kindle Edition by Imran Bashir

**BTBC 610-23 Cyber Security in Blockchain Technology Lab 0L 0T 2P 1C**

Experiment based upon the theory subject.