Scheme & Syllabus of

Bachelor ofTechnology

Computer Science & Engineering

Batch 2018 onwards

By

Board of Study CSE on 27 April 2018

Department of Academics

IK GujralPunjab Technical University

Bachelor of Technology in Computer Science & Engineering

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

Courses & Examination Scheme:

Third Semester

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

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	1	0	40	60	100	3
EVS102	Mandatory Courses	Environmental Sciences	-	-	-	-	-	-	0
BTES 402-18	Engineering ScienceCourse	Computer Organization & Architecture Lab	0	0	2	30	20	50	1
BTCS 404-18	Professional CoreCourses	Operating Systems Lab	0	0	4	30	20	50	2
BTCS 405-18	Professional CoreCourses	Design & Analysis of Algorithms Lab	0	0	4	30	20	50	2
Total			**14**	**2**	**10**	**290**	**360**	**650**	**24**

Fifth Semester

Course Code	Type of Course	Course Title	Hours per Week			Marks Distribution		Total Marks	Credits
			L	T	P	Internal	External		
BTES 501-18	Engineering Science	Signals & Systems	3	0	0	40	60	100	3
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 503-18	Professional Core Courses	Software Engineering	3	0	0	40	60	100	3
BTHU 501-18	Humanities & Social Sciences including Management Courses	Humanities-II	3	0	0	40	60	100	3
BTCS XXX-18	Professional Elective	Elective-I	3	0	0	40	60	100	3
MC	Mandatory Courses	Constitution of India/ Essence of IndianTraditional Knowledge	-	-	-	-	-	-	0
BTCS 504-18	Professional Core Courses	Database Management Systems Lab	0	0	4	30	20	50	2
BTCS 505-18	Professional Core Courses	Software Engineering Lab	0	0	4	30	20	50	2
Total			**18**	**0**	**8**	**300**	**400**	**700**	**22**

Sixth 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	Computer Networks	3	0	0	40	60	100	3
BTCS YYY-18	Professional Elective Courses	Elective-II	3	0	0	40	60	100	3
BTCS ZZZ-18	Professional Elective Courses	Elective-III	3	0	0	40	60	100	3
BTOE ***	Open Elective Courses	Open Elective-I (Humanities)	3	0	0	40	60	100	3
BTCS 603-18	Project	Project-1	0	0	6	60	40	100	3
BTCS 604-18	Professional Core Courses	Compiler Design Lab	0	0	4	30	20	50	2
BTCS 605-18	Professional Core Courses	Computer Networks Lab	0	0	4	30	20	50	2
Total			**15**	**0**	**14**	**320**	**380**	**700**	**22**

Seventh Semester

Course Code	Type of Course	Course Title	Hours per Week			Marks Distribution		Total Marks	Credits
			L	T	P	Internal	External		
BTCS UUU-18	Professional Elective Courses	Elective-IV	3	0	0	40	60	100	3
BTCS VVV-18	Professional Elective	Elective-V	3	0	0	40	60	100	3
BTOE ***	Open Elective Courses	Open Elective-II	3	0	0	40	60	100	3
BTCS 701-18	Professional Core Courses	Artificial Intelligence	3	0	0	40	60	100	3
BTCS 702-18	Project	Project-II	0	0	12	100	50	150	6
BTCS 703-18	Professional Training	Industrial *Training	-	-	-	60	40	100	2
Total			**12**	**0**	**12**	**320**	**330**	**650**	**20**

* The students will take 6-8 weeks summer training in Industry after semester 6th.

Eighth Semester

Course Code	Type of Course	Course Title	Hours per Week			Marks Distribution		Total Marks	Credits
			L	T	P	Internal	External		
BTCS XXX-18	Professional Elective Courses	Elective-VI	3	0	0	40	60	100	3
BTOE * **	Open Elective Courses	Open Elective-III	3	0	0	40	60	100	3
BTOE * **	Open Elective Courses	Open Elective-IV	3	0	0	40	60	100	3
BTCS 801-18	Project	Project-III	0	0	12	100	50	150	6
Total			**9**	**0**	**12**	**220**	**230**	**450**	**15**

LIST OF ELECTIVES

Elective-I

BTCS 506-18 Programming in Java
BTCS 507-18 Web Technologies
BTCS 508-18 Computational Biology
BTCS 509-18 Computer Graphics

Elective-II

BTCS 606-18 Mobile Application Development
BTCS 607-18 Distributed Systems
BTCS 608-18 Machine Learning
BTCS 609-18 Digital Signal Processing

Elective-III

BTCS 610-18 Parallel and Distributed Algorithms
BTCS 611-18 Embedded Systems
BTCS 612-18 Data Mining
BTCS 613-18 Cloud Computing

Elective-IV

BTCS 702-18 Information Theory and Coding
BTCS 703-18 Distributed Operating System
BTCS 704-18 Soft Computing
BTCS 705-18 Human Computer Interaction

Elective-V

BTCS 706-18 Computational Number Theory
BTCS 707-18 Ad-Hoc and Sensor Networks
BTCS 708-18 Speech and Natural Language Processing
BTCS 709-18 Parallel Architectures

Elective-VI

BTCS 802-18 Queuing Theory and Modeling
BTCS 803-18 Real Time Systems
BTCS 804-18 Data Analytics
BTCS 805-18 Image Processing

LIST OF OPEN ELECTIVES

Open Elective-I

1. Soft skills and interpersonal communication
2. Cyber law and ethics
3. Human resource development and organizational behavior
4. Economics policies in India

Open electives offered by the department:

Courses of odd semesters:

BTCS301-18 Data Structures & Algorithms

BTCS503-18 Object Oritented Programming

BTCS501-18 Database Management System

Courses of even semesters:

BTES401-18 Computer organisation & Arcitecture

BTCS402-18 Operating system

BTCS602-18 Computer Networks

LIST OF COURSES FOR HONOURS DEGREE

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

Course Code	Type of courses	Course Title	Hours per Week			Marks Distributions		Total Marks	Credits
			L	T	P	Internal	External		
BTCS H01-18	Professional Elective Courses	Graph Theory	3	0	0	40	60	100	3
BTCS H02-18	Professional Elective Courses	Computer Vision	3	0	0	40	60	100	3
BTCS 611-18	Professional Elective Courses	Embedded Systems	3	0	0	40	60	100	3
BTCS H03-18	Professional Elective Courses	Software Project Management	3	0	0	40	60	100	3
BTCS H04-18	Professional Elective Courses	Cryptography & Network Security	3	0	0	40	60	100	3
BTCS H05-18	Professional Elective Courses	Internet-of-Things	3	0	0	40	60	100	3
BTCS 804-18	Professional Elective Courses	Data Analytics	3	0	0	40	60	100	3
BTCS 608-18	Professional Elective Courses	Machine Learning	3	0	0	40	60	100	3
BTCS H06-18	Professional Elective Courses	ICT in Agriculture and Rural Development	3	0	0	40	60	100	3
BTCS H07-18	Professional Elective Courses	Computational Technologies for Smart Cities	3	0	0	40	60	100	3
BTCS H08-18	Professional Elective Courses	Computer Forensics	3	0	0	40	60	100	3

BTCS H09-18	Professional Elective Courses	Advavanced Algorithms	3	0	0	40	60	100	3

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.

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

Detailed Contents:

Module 1:

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

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

Module 2:

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

Module 3:

Exact, linear and Bernoulli's equations, Euler's equations, Equations not of first degree: equations solvable for p, equations solvable for y, equations solvable for x and Clairaut's type.

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

Module 4:

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

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

1.Understand the functions of several variables that are essential in mostbranches of engineering;

2. Apply multiple integrals to deal with areas and volumes of various structures which are quite significant in real world;

3. Formulate and solveengineering problems related to convergence, infinite series, power series and Taylor series;

4. Create, select and utilize the learnt techniques of first degree ordinary differential equationsto model real world problems &;

5. Be acquainted with the knowledge required to solve higher order ordinary differential equations.

Textbooks/References:

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

Development of Societies
Course code: HSMC101-18

Credits: 3

COURSE TOPICS:

2.1 Unit I: Social Development (5 hours)

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

2.2 Unit II: Political Development (3 hours)

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

2.3 Unit III: Economic Development (18 hours)

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

Gandhian idea of development. Swaraj and Decentralization.

3. READINGS

3.1 TEXTBOOK:

3.2 *REFERENCE BOOKS:

4. OTHER SESSIONS

4.1 *TUTORIALS:

4.2 *LABORATORY:

4.3 *PROJECT: Possible projects in this course could be

a) Interact with local communities and understand their issues.

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

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

PHILOSOPHY
Course code: HSMC102-18

Credits: 3

COURSE TOPICS:

2.1 Unit 1:
The difference between knowledge (Vidya) and Ignorance (Avidya):
a. Upanishads;
b. Six systems orthodox and Heterodox Schools of 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.