



# AMRITA SAI INSTITUTE OF SCIENCE AND TECHNOLOGY (AUTONOMOUS)

Approved by AICTE, New Delhi; Permanently Affiliated To JNTU, Kakinada

ACCREDITED BY NAAC WITH  A  GRADE

PARITALA(P), KANCHIKACHERLA(M), KRISHNA (D)  521 180(A. P.)



## DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

### COURSE STRUCTURE FOR CSE/CBA/CSD

CSE : Computer Science and Engineering

CBA: Big Data Analytics

CSD: Data Science

**AR20 REGULATION**

#### I YEAR I SEMESTER:

SNO	SUB CODE	COURSE TITLE	CATEGORY	L	T	P	CREDITS
1	20HS1T1	COMMUNICATIVE ENGLISH-I	HSMC	3	0	0	3
2	20BS1T2	ORDINARY DIFFERENTIAL EQUATIONS AND CALCULUS	BSC	3	0	0	3
3	20BS1T3	ENGINEERING CHEMISTRY	BSC	3	0	0	3
4	20CS1T4	IT WORKSHOP	ESC	1	0	4	3
5	20CS1T5	PYTHON <input type="checkbox"/> I	ESC	3	0	0	3
6	20HS1L1	COMMUNICATIVE ENGLISH LAB	HSMC	0	0	3	1.5
7	20BS1L2	ENGINEERING CHEMISTRY LAB	BSC	0	0	3	1.5
8	20CS1L3	PYTHON LAB-I	ESC	0	0	3	1.5
<b>TOTAL CREDITS</b>							<b>19.5</b>

#### I YEAR II SEMESTER:

S.NO	SUB CODE	COURSE TITLE	CATEGORY	L	T	P	CREDITS
1	20BS2T1	APPLIED PHYSICS	BSC	3	0	0	3
2	20BS2T2	LINEAR ALGEBRA AND TRANSFORMATIONS	BSC	3	0	0	3
3	20CS2T3	DATA STRUCTURES USING C++	ESC	3	0	0	3
4	20CS2T4	PYTHON <input type="checkbox"/> II	ESC	3	0	0	3
5	20CS2T5	ENGINEERING GRAPHICS AND DESIGN	ESC	1	0	4	3
6	20BS2L1	APPLIED PHYSICS LAB	BSC	0	0	3	1.5
7	20CS2L2	DATA STRUCTURES AND C++ LAB	BSC	0	0	3	1.5
8	20CS2L3	PYTHON LAB-II	ESC	0	0	3	1.5
9	20MC2T6	ENVIRONMENTAL STUDIES	MC	<b>2</b>	<b>0</b>	<b>0</b>	<b>0</b>

**TOTAL CREDITS****19.5****II YEAR I SEMESTER:**

SNO	COURSE CODE	COURSE TITLE	CATEGORY	L	T	P	CREDITS
1	20BS3T1	Probability and Statistics	BS	3	0	0	3
2	20CS3T2	Unix and Shell Programming	CS	3	0	0	3
3	20CS3T3	Advanced Data Structures and Algorithms	CS	3	0	0	3
4	20CS3T4	Java Programming	CS	3	0	0	3
5	20CS3T5	Digital Logic Design	ES	3	0	0	3
6	20CS3L1	ADS&A Lab	CS	3	0	0	1.5
7	20CS3L2	Java Programming Lab	CS	0	0	3	1.5
8	20CS3L3	Unix and Shell Programming Lab	CS	0	0	3	1.5
9	20SE3T1	Introduction to Computing Principles	SE	2	0	1	2
10	20MC3T1	Essence of Indian Traditional Knowledge	MC	2	0	0	0
<b>TOTAL CREDITS</b>							<b>21.5</b>

**II YEAR II SEMESTER:**

SNO		COURSE TITLE	CATEGORY	L	T	P	CREDITS
1	20CS4T1	PHP programming	CS	3	0	0	3
2	20CS4T2	Computer Organization	CS	3	0	0	3
3	20CS4T3	Advanced Java Programming	CS	3	0	0	3
4	20CB4T1	Statistics with R Programming	CS	3	0	0	3
5	20HS4T1	Communicative English II	HS	3	0	0	3
6	20CS4L1	PHP programming Lab	CS LAB	0	0	3	1.5
7	20CS4L2	Advanced Java Lab	CS LAB	0	0	3	1.5
8	20CB4L1	R Programming Lab	CS LAB	0	0	3	1.5
9	20SE4T1	Solving Problem with Computers	SE	1	0	2	2
<b>TOTAL CREDITS</b>							<b>21.5</b>
<b>Internship 2 Months (Mandatory) during summer vacation</b>							

### III YEAR I SEMESTER:

SNO	Course Code	COURSE TITLE	CATEGORY	L	T	P	CREDITS
1	20CS5T1	Web Technologies	CS	3	0	0	3
2	20CS5T2	Database Management Systems	CS	3	0	0	3
3	20CS5T3	Computer Networks	CS	3	0	0	3
4	20CS5T4	Elective-I	EL	3	0	0	3
5	20CS5T5	Open Elective-I	OE	2	0	2	3
6	20CS5L1	DBMS Lab	CS LAB	0	0	3	1.5
7	20CS5L2	Web Technologies Lab	CS LAB	0	0	3	1.5
8	20SE5L1	Adv English Communication skills lab-I	SE	1	0	2	2
9	20MC5T1	Professional Ethics & Human Values	MC	2	0	0	0
	Summer Internship 2 Months (Mandatory) after second year (to be evaluated during V semester)			0	0	0	1.5
<b>TOTAL CREDITS</b>							<b>21.5</b>

### III YEAR II SEMESTER:

SNO		COURSE TITLE	CATEGORY	L	T	P	CREDITS
1	20CB6T1	Data Warehousing and Data Mining	CS	3	0	0	3
2	20CS6T2	Design and Analysis of Algorithms	CS	3	0	0	3
3	20CS6T3	Object Oriented Analysis and Design using UML	CS	3	0	0	3
4		Elective-II	CS	2	0	2	3
5		Open Elective-II	CS	3	0	0	3
6	20CS6L1	Object Oriented Analysis and Design using UML Lab	CS LAB	0	0	3	1.5
7	20CB6L1	Data Warehousing and Data Mining lab	CS LAB	0	0	3	1.5
8	20CS6L3	Computer Networks Lab	CS LAB	0	0	3	1.5
9	20SE6T1	Adv English Communication skills Lab-II	SE	1	0	2	2
10	20MC6T1	Employability Skills	MC	2	0	0	0
<b>TOTAL CREDITS</b>							<b>21.5</b>
<b>Industrial/Research Internship (Mandatory) 2 Months during summer vacation</b>							

### IV YEAR I SEMESTER:

SNO		COURSE TITLE	CATEGORY	L	T	P	CREDITS
1	20CS7T1	Cryptography and Network Security	CS	3	0	0	3
2	20CB7T1	Hadoop and Big Data	CS	3	0	0	3
3	20CS7T3	Artificial Intelligence	CS	3	0	0	3
4		Open Elective-III	EL	2	0	2	3
5		Open Elective-IV	EL	2	0	2	3
6	<ul style="list-style-type: none"> <li>• 20HS7T6A</li> <li>• 20HS7T6B</li> <li>• 20HS7T6C</li> <li>20HS7T6D</li> </ul>	<ul style="list-style-type: none"> <li>• Principles of Microeconomics</li> <li>• International Relations</li> <li>• Social Problems</li> <li>• Southern Culture</li> </ul>	HS	3	0	0	3
7	20SE7T1	Industrial Automation	SE	1	0	2	2
Industrial/Research Internship 2 Months (Mandatory) after third year (to be evaluated during VII semester)				0	0	0	3
<b>TOTAL CREDITS</b>							<b>23</b>

**IV YEAR II SEMESTER:**

SNO	CODE	CATEGORY	L	T	P	CREDITS
1	20CSP1	Major Project	0	0	0	12
<b>Internship (6 MONTHS)</b>						
<b>TOTAL CREDITS</b>						<b>12</b>

**ELECTIVE-I:**

- 20CS5T4A Data Science
- 20CS5T4B Big Data Analytics
- 20CS5T4C Artificial Intelligence
- 20EC5T4D Miro Processor and Micro Controller
- 

**ELECTIVE-II:**

- 20CS6T4A Computer Graphics
- 20CS6T4B Compiler Design
- 20CS6T4C Machine Learning
- 20CS6T4D Neural networks and fuzzy logic

**OPEN ELECTIVE – I:**

- 20CS5T5A Big Data Analytics/Data Science I
- 20CS5T5B IoT and Cyber security with Blockchain Tech I
- 20CS5T5C Artificial Intelligence and Machine Learning I
- 20CS5T5D Cloud Computing I

**OPEN ELECTIVE-II:**

- 20CS6T5A Big Data Analytics/Data Science II
- 20CS6T5B IoT and Cyber security with Blockchain Tech II
- 20CS6T5C Artificial Intelligence and Machine Learning II
- 20CS6T5D Cloud Computing II

**OPEN ELECTIVE-III:**

- 20CS7T4A Big Data Analytics/Data Science III
- 20CS7T4B IoT and Cyber security with Blockchain Tech III
- 20CS7T4C Artificial Intelligence and Machine Learning III
- 20CS7T4D Cloud Computing III

**OPEN ELECTIVE-IV:**

- 20CS7T5A Big Data Analytics/Data Science IV
- 20CS7T5B IoT and Cyber security with Blockchain Tech IV
- 20CS7T5C Artificial Intelligence and Machine Learning IV
- 20CS7T5D Cloud Computing IV



**Amrita Sai Institute of Science and Technology  
(Autonomous)**

Approved by AICTE, New Delhi; Permanently Affiliated to JNTUK, Kakinada  
ISO 9001:2015 Certified Institution, Accredited by NAAC with 'A' grade  
Recognized by UGC under 2(f) and 12(B) of 1956 Act  
Amrita Sai Nagar, Paritala, Krishna District, Andhra Pradesh-521180  
[www.amritasai.org.in](http://www.amritasai.org.in), 0866-2428399



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

Course Code	<b>PYTHON-I</b>	<b>T</b>	<b>P</b>
20CS1T5		<b>4</b>	<b>3</b>
<b>I Year I Semester</b>			
<b>COURSE OBJECTIVES</b>			
1	To acquire programming skills in core Python.		
2	To acquire Object Oriented Skills in Python		
3	To develop the skill of designing Graphical user Interfaces in Python		
	To develop the ability to write database applications in Python		
<b>COURSE OUTCOMES</b>			
1	Explain basic principles of Python programming language		
2	Implement object oriented concepts,		
3	Implement database and GUI applications.		

**UNIT – I:**

**Introduction:** History of Python, Need of Python Programming, Applications Basics of Python Programming Using the REPL(Shell), Running Python Scripts, Variables, Assignment, Keywords, Input-Output, Indentation.

**UNIT – II:**

**Types, Operators and Expressions:** Types - Integers, Strings, Booleans; Operators- Arithmetic Operators, Comparison (Relational) Operators, Assignment Operators, Logical Operators, Bitwise Operators, Membership Operators, Identity Operators,

**UNIT-III**

Expressions and order of evaluations Control Flow- if, if-elif-else, for, while, break, continue, pass

**UNIT – IV**

**Data Structures** Lists - Operations, Slicing, Methods; Tuples, Sets, Dictionaries, Sequences. Comprehensions.

**UNIT – V:**

**Functions** - Defining Functions, Calling Functions, Passing Arguments, Keyword

Arguments, Default Arguments, Variable-length arguments,

**UNIT-VI:**

Anonymous Functions, Fruitful Functions(Function Returning Values) Python lambda.

**TEXT BOOKS**

1. Python Programming: A Modern Approach, Vamsi Kurama, Pearson
2. Learning Python, Mark Lutz, Orielly

**Reference Books:**

1. Think Python, Allen Downey, Green Tea Press
2. Core Python Programming, W.Chun, Pearson.
3. Introduction to Python, Kenneth A. Lambert, Cengage




**Amrita Sai Institute of Science and Technology  
(Autonomous)**

Approved by AICTE, New Delhi; Permanently Affiliated to JNTUK, Kakinada  
ISO 9001:2015 Certified Institution, Accredited by NAAC with 'A' grade  
Recognized by UGC under 2(f) and 12(B) of 1956 Act  
Amrita Sai Nagar, Paritala, Krishna District, Andhra Pradesh-521180  
[www.amritasai.org.in](http://www.amritasai.org.in), 0866-2428399



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

Course Code 20CS1T4	<b>IT WORKSHOP</b>				<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
	Maximum expected contact hours : <b>60</b>				<b>1</b>	<b>0</b>	<b>4</b>	<b>3</b>
<b>PURPOSE:</b> This course aims at study of is the use of computers to store, retrieve, transmit, and manipulate data or information, often in the context of a business or other enterprise.								
<b>INSTRUCTIONAL COURSE OBJECTIVES</b>								
1	Understand the basic components and peripherals of a computer							
2	To become familiar in configuring a system							
3	Learn the usage of productivity tools.							
4	Acquire knowledge about the netiquette and cyber hygiene							
5	Get hands on experience in trouble shooting a system?							
<b>COURSE OUTCOMES</b>								
1	Students are able to understanding of concepts, patterns of decentralization implementation in Africa							
2	Students are able to Identified opportunities for coordinated policy responses, capacity building and implementation of best practices							

**UNIT -1** Introduction to Computers: Generation of computers, Characteristic and classifications of computers. Components of Computer: CPU, Various I/O Devices, Memory & its types , (Memory Hierarchy, Storage Media), Computer Software and their types, Operating System.

**UNIT-II** Computer Networks & Communication: LAN, MAN, WAN, Network Topologies, Modes of Data Communication. Introduction to Internet and its Safeguard: Internet Addresses, Domain Name System, URL, Web Browsers Search Engines, Firewalls, Anti-Virus, Translators.

**UNIT-III**

Number system: Binary number system, Decimal number system, Octal number system, Hexadecimal number systems, conversion decimal to binary, binary to decimal, octal to binary etc, binary addition, binary subtraction – One’s complement Two’s complement method, binary multiplication, binary division Concept of Information Systems and Software : Information gathering

#### UNIT-IV

Concept of Information Systems and Software: Information gathering, requirement and feasibility analysis, data flow diagrams, process specifications, input/output design, process life cycle, planning and managing the project

#### UNIT-V

Ms. Word: Creating, editing, saving and printing text document, Font and paragraph formatting, Simple character formatting, Inserting tables, smart art, page breaks Using lists and styles, working with images, Using Spelling and Grammar check Understanding document properties, Mail Merge

#### UNIT-VI

Ms. Power Point: Opening, viewing, creating, and printing slide, Applying auto layouts, Adding custom animation Using slide transitions, graphically representing data: Charts & Graphs, Creating Professional Slide for Presentation

**Text Books:**

1. *Computer Hardware, Installation, Interfacing, Troubleshooting and Maintenance*, K.L. James, Eastern Economy Edition.
2. *Microsoft Office 2007: Introductory Concepts and Techniques, Windows XP Edition* by Gary B. Shelly, Misty E. Vermaat and Thomas J. Cashman (2007, Paperback).
3. *Introduction to Information Technology*, ITL Education Solutions limited, Pearson Education.






**Amrita Sai Institute of Science and Technology  
(Autonomous)**

Approved by AICTE, New Delhi; Permanently Affiliated to JNTUK, Kakinada  
ISO 9001:2015 Certified Institution, Accredited by NAAC with 'A' grade  
Recognized by UGC under 2(f) and 12(B) of 1956 Act  
Amrita Sai Nagar, Paritala, Krishna District, Andhra Pradesh-521180  
[www.amritasai.org.in](http://www.amritasai.org.in), 0866-2428399



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Course Code 20CS1L3	<b>PYTHON LAB-I</b>	<b>T</b>	<b>P</b>
		-	<b>3</b>
<b>I Year I Semester</b>			
<b>COURSE OBJECTIVES</b>			
1	To be able to introduce core programming basics and program design with functions using Python programming language.		
2	To understand a range of Object-Oriented Programming, as well as in-depth data and information processing techniques.		
3	To understand the high-performance programs designed to strengthen the practical expertise.		
<b>COURSE OUTCOMES</b>			
1	Student should be able to understand the basic concepts scripting and the contributions of scripting language		
2	Ability to explore python especially the object oriented concepts, and the built in objects of Python.		
3	Ability to create practical and contemporary applications such as TCP/IP network programming, Web applications, discrete event simulations		

**Exercise 1 - Basics**

- a) Running instructions in Interactive interpreter and a Python Script
- b) Write a program to purposefully raise Indentation Error and Correct it

**Exercise 2 - Operations**

- a) Write a program to compute distance between two points taking input from the user (Pythagorean Theorem)
- b) Write a program add.py that takes 2 numbers as command line arguments and prints its sum.

**Exercise - 3 Control Flow-I**

- a) Write a Program for checking whether the given number is a even number or not.
- b) Using a for loop, write a program that prints out the decimal equivalents of  $1/2, 1/3, 1/4, \dots, 1/10$

**Exercise - 4 Control Flow-II**

- c) Write a program using a for loop that loops over a sequence. What is sequence ?
- d) Write a program using a while loop that asks the user for a number, and prints a countdown from that number to zero.

**Exercise 5 - Control Flow - Continued**

- a) Find the sum of all the primes below two million.  
Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting with 1 and 2, the first 10 terms will be:

1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

- b) By considering the terms in the Fibonacci sequence whose values do not exceed four million, find the sum of the even-valued terms.

**Exercise - 6 - DS**

- a) Write a program to count the numbers of characters in the string and store them in a dictionary data structure
- b) Write a program to use split and join methods in the string and trace a birthday with a dictionary data structure.

**Exercise - 7 DS - Continued**

- a) Write a program combine\_lists that combines these lists into a dictionary.
- b) Write a program to count frequency of characters in a given file. Can you use character frequency to tell whether the given file is a Python program file, C program file or a text file?

**Exercise - 8 Functions**

- a) Write a function ball\_collide that takes two balls as parameters and computes if they are colliding. Your function should return a Boolean representing whether or not the balls are colliding.

Hint: Represent a ball on a plane as a tuple of (x, y, r), r being the radius  
If (distance between two balls centers)  $\leq$  (sum of their radii) then (they are colliding)

- b) Find mean, median, mode for the given set of numbers in a list.

**Exercise - 9 Functions - Continued**

- a) Write a function nearly\_equal to test whether two strings are nearly equal. Two strings a and b are nearly equal when a can be generated by a single mutation on b.
- b) Write a function dups to find all duplicates in the list.
- c) Write a function unique to find all the unique elements of a list.

**Exercise - 10 - Functions - Problem Solving**

- a) Write a function cumulative\_product to compute cumulative product of a list of numbers.
- b) Write a function reverse to reverse a list. Without using the reverse function.
- c) Write function to compute gcd, lcm of two numbers. Each function shouldn't exceed one line.




**Amrita Sai Institute of Science and Technology  
(Autonomous)**

Approved by AICTE, New Delhi; Permanently Affiliated to JNTUK, Kakinada  
ISO 9001:2015 Certified Institution, Accredited by NAAC with 'A' grade  
Recognized by UGC under 2(f) and 12(B) of 1956 Act  
Amrita Sai Nagar, Paritala, Krishna District, Andhra Pradesh-521180  
[www.amritasai.org.in](http://www.amritasai.org.in), 0866-2428399



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

Course Code <b>20CS2T4</b>	<b>PYTHON-II</b>	<b>T</b>	<b>P</b>
		<b>4</b>	<b>3</b>
<b>I Year II Semester</b>			
<b>COURSE OBJECTIVES</b>			
1	To acquire programming skills in core Python.		
2	To acquire Object Oriented Skills in Python		
3	To develop the skill of designing Graphical user Interfaces in Python		
	To develop the ability to write database applications in Python		
<b>COURSE OUTCOMES</b>			
1	Explain basic principles of Python programming language		
2	Implement object oriented concepts,		
3	Implement database and GUI applications.		

**UNIT-I**

Scope of the Variables in a Function Global and Local Variables. **Modules:** Creating modules, import statement, from. Import statement, name spacing,

**Python packages**, Introduction to PIP, Installing Packages via PIP, Using Python Packages

**UNIT-II**

**Object Oriented Programming OOP in Python:** Classes, 'self variable', Methods, Constructor Method, ,Inheritance, Overriding Methods, Data hiding,

**UNIT-III**

**Python Arrays**, Iterator, multi D-lists ,scope,module,date,math,Json

**Python file handling**, Read files, create files, delete files

**UNIT –IV**

**Error and Exceptions:** Difference between an error and Exception, Handling Exception, try except block, Raising Exceptions, User Defined Exceptions

**UNIT –V:**

**Brief Tour of the Standard Library** - Operating System Interface - String Pattern Matching, Mathematics, Internet Access, Dates and Times, Data Compression, Multithreading, GUI Programming, Turtle Graphics

**UNIT-VI:**

**Testing:** Why testing is required ?, Basic concepts of testing, Unit testing in Python, Writing Test cases, Running Tests.

**TEXT BOOKS**

1. Python Programming: A Modern Approach, Vamsi Kurama, Pearson
2. Learning Python, Mark Lutz, Orielly

**Reference Books:**

1. Think Python, Allen Downey, Green Tea Press
2. Core Python Programming, W.Chun, Pearson.
3. Introduction to Python, Kenneth A. Lambert, Cengage






**Amrita Sai Institute of Science and Technology  
(Autonomous)**

Approved by AICTE, New Delhi; Permanently Affiliated to JNTUK, Kakinada  
ISO 9001:2015 Certified Institution, Accredited by NAAC with 'A' grade  
Recognized by UGC under 2(f) and 12(B) of 1956 Act  
Amrita Sai Nagar, Paritala, Krishna District, Andhra Pradesh-521180  
[www.amritasai.org.in](http://www.amritasai.org.in), 0866-2428399



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

<b>Course Code</b>	<b>B.Tech IYear II Sem</b>	<b>DATA STRUCTURES USING C++</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
<b>20CS2T3</b>			<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

<i>Course Code</i> 20CS2T3	<i>DATA STRUCTURES USING C++</i>	<i>L</i>	<i>T</i>	<i>P</i>	<i>C</i>
		<b>3</b>	<b>1</b>	<b>--</b>	<b>3</b>
<b>PURPOSE:</b> <i>Data structure is designed to organize data to suit a specific purpose so that it can be accessed and worked with in appropriate ways.</i>					
<b>INSTRUCTIONAL COURSE OBJECTIVES</b>					
<i>1</i>	<i>To be familiar with basic techniques of object oriented principles and exception handling using C++</i>				
<i>2</i>	<i>To be familiar with the concepts like Inheritance, Polymorphism</i>				
<i>3</i>	<i>Solve problems using data structures such as linear lists, stacks, queues, hash tables</i>				
<i>4</i>	<i>Be familiar with advanced data structures such as balanced search trees, AVL Trees, and B Trees.</i>				
<b>COURSE OUTCOMES</b>					
<i>1</i>	<i>Understand the basic terminology used in computer programming</i>				
<i>2</i>	<i>Write, compile and debug programs in C language.</i>				
<i>3</i>	<i>Use different data types in a computer program.</i>				
<i>4</i>	<i>Design programs involving decision structures, loops and functions.</i>				
<i>5</i>	<i>Explain the difference between call by value and call by reference</i>				
<i>6</i>	<i>Understand the dynamics of memory by the use of pointers</i>				

**UNIT-I: ARRAYS**

*Abstract Data Types and the C++ Class, An Introduction to C++ Class- Data Abstraction and Encapsulation in C++- Declaring Class Objects and Invoking Member Functions-*

*The Array as an Abstract Data Type, The Polynomial Abstract Data type- Polynomial Representation- Polynomial Addition. Sparse Matrices, Introduction- Sparse Matrix Representation- Transposing a Matrix- Matrix Multiplication,*

**UNIT-II: STACKS AND QUEUES**

*Templates in C++, Template Functions- Using Templates to Represent Container Classes, The Stack Abstract Data Type, The Queue Abstract Data Type, Evaluation of Expressions, Expression- Postfix Notation- Infix to Postfix, Infix to prefix and prefix evaluation*

### **UNIT-III: LINKED LISTS**

*Single Linked List and Chains, Representing Chains in C++, Defining a Node in C++- Pointer manipulation in C++- Chain Manipulation Operations, The Template Class Chain, Implementing Chains with Templates- Chain Iterators- Chain Operations- Reusing a Class, Circular Lists, Space Lists, Linked Stacks and Queues, Polynomial Representation- Adding Polynomials-, Equivalence Classes, Sparse Matrices, Sparse Matrix Representation.*

### **UNIT-IV: TREES**

*Introduction, Terminology, Representation of Trees, Binary Trees, The Abstract Data Type, Properties of Binary Trees, Binary Tree Representations, Binary Tree Traversal and Tree Iterators, Introduction, In order Traversal Preorder Traversal, Post order Traversal, Tree traversals using stack(non recursive), Thread Binary Trees, Threads, In order Traversal of a Threaded Binary Tree, Inserting a Node into a Threaded Binary Tree, Heaps, Priority Queues, Definition of a Max Heap, Insertion into a Max Heap, Deletion from a Max Heap, Binary Search Trees, Definition, Searching a Binary Search Tree, Insertion into a Binary Search Tree, Deletion from a Binary Search Tree, Height of Binary Search Tree.*

### **UNIT-V: GRAPHS**

*The Graph Abstract Data Type, Introduction, Definition, Graph Representation, Elementary Graph Operation, Depth First Search, Breadth First Search, Connected Components, Spanning Trees, , Minimum Cost Spanning Trees, Kruskal S Algorithm, Prim s Algorithm.*

### **UNIT-VI: SORTING**

*Insertion Sort, Divide and conquer strategy, Quick Sort, Merge Sort Merging, Iterative Merge Sort, Recursive Merge Sort, Heap Sort. Radix sort*

#### **TEXT BOOKS:**

1. *Data structures, Algorithms and Applications in C++, S.Sahni, University Press (India) Pvt.Ltd, 2<sup>nd</sup> ed*
2. *Data structures and Algorithm Analysis in C++, Mark Allen Weiss, Pearson Education. Ltd., 2<sup>nd</sup> ed*
3. *Data structures and Algorithms in C++, Michael T.Goodrich, R.Tamassia and .Mount, Wiley student ed*

**REFERENCE BOOKS:**

1. *Data structures and algorithms in C++, 3rd Edition, Adam Drozdek, Thomson*
2. *Data structures using C and C++, Langsam, Augenstein and Tanenbaum, PHI. Problem solving with C++, The OOP, Fourth edition, W.Savitch, Pearson education.*






**Amrita Sai Institute of Science and Technology  
(Autonomous)**

Approved by AICTE, New Delhi; Permanently Affiliated to JNTUK, Kakinada  
ISO 9001:2015 Certified Institution, Accredited by NAAC with 'A' grade  
Recognized by UGC under 2(f) and 12(B) of 1956 Act  
Amrita Sai Nagar, Paritala, Krishna District, Andhra Pradesh-521180  
[www.amritasai.org.in](http://www.amritasai.org.in), 0866-2428399



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Course Code 20CS2L3	<b>PYTHON LAB-II</b>	<b>T</b>	<b>P</b>
		-	3
<b>I Year II Semester</b>			
<b>COURSE OBJECTIVES</b>			
1	To be able to introduce core programming basics and program design with functions using Python programming language.		
2	To understand a range of Object-Oriented Programming, as well as in-depth data and information processing techniques.		
3	To understand the high-performance programs designed to strengthen the practical expertise.		
<b>COURSE OUTCOMES</b>			
1	Student should be able to understand the basic concepts scripting and the contributions of scripting language		
2	Ability to explore python especially the object oriented concepts, and the built in objects of Python.		
3	Ability to create practical and contemporary applications such as TCP/IP network programming, Web applications, discrete event simulations		

**Exercise - 1 Files**

- a) Write a program to print each line of a file in reverse order.
- b) Write a program to compute the number of characters, words and lines in a file.

**Exercise 2 - Multi-D Lists**

- a) Write a program that defines a matrix and prints
- b) Write a program to perform addition of two square matrices

**Exercise 3 - Multi-D Lists continued**

- a) Write a program to perform multiplication of two square matrices

**Exercise - 4 - Modules**

- a) Install packages requests, flask and explore them. using (pip)
- b) Write a script that imports requests and fetch content from the page. Eg. (Wiki)
- c) Write a simple script that serves a simple HTTPResponse and a simple HTML Page

**Exercise - 5 OOP**

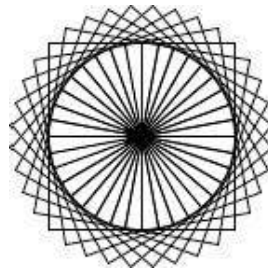
- a) Class variables and instance variable and illustration of the self variable
  - i) Robot
  - ii) ATM Machine

**Exercise - 6 GUI**

- a) Write a GUI for an Expression Calculator using tk

**Exercise - 7 Graphics**

- a) Write a program to implement the following figures using turtle



**Exercise - 8 - Testing**

- a) Write a test-case to check the function `even_numbers` which return True on passing a list of all even numbers
- b) Write a test-case to check the function `reverse_string` which returns the reversed string

**Exercise -9 - Advanced**

- a) Build any one classical data structure.
- b) Write a program to solve knapsack problem




**Amrita Sai Institute of Science and Technology  
(Autonomous)**

Approved by AICTE, New Delhi; Permanently Affiliated to JNTUK, Kakinada  
ISO 9001:2015 Certified Institution, Accredited by NAAC with 'A' grade  
Recognized by UGC under 2(f) and 12(B) of 1956 Act  
Amrita Sai Nagar, Paritala, Krishna District, Andhra Pradesh-521180  
[www.amritasai.org.in](http://www.amritasai.org.in), 0866-2428399



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Course Code	<b>B.Tech IYear II Sem</b>	<b>DATA STRUCTURES USING C++ LAB</b>	L	T	P	C
<b>20CS2L2</b>			--	--	4	1.5

**PURPOSE:** Data structure is designed to organize data to suit a specific purpose so that it can be accessed and worked with in appropriate ways

**INSTRUCTIONAL COURSE OBJECTIVES**

- |   |  |
|---|--|
| 1 | To develop skills to design and analyze simple linear and non linear data structures                         |
| 2 | To Strengthen the ability to identify and apply the suitable data structure for the given real world problem |
| 3 | To Gain knowledge in practical applications of data structures   |

**COURSE OUTCOMES**

- |   |   |
|---|---|
| 1 | Be able to design and analyze the time and space efficiency of the data structure |
| 2 | Be capable to identify the appropriate data structure for given problem           |
| 3 | Have practical knowledge on the application of data structures                    |

1. Implementation of Singly linked list with all operations
2. Implementation of Multistack in a Single Array.
3. Implementation of Circular Queue.
4. Implementation of Binary Search trees.
5. Implementation of Hash table.
6. Implementation of Heaps.
7. Implementation of Breadth First Search Techniques.
8. Implementation of Depth First Search Techniques.

9. *Implementation of Prim's Algorithm.*

10. *Implementation of Dijkstra's Algorithm.*

11. *Implementation of Kruskal's Algorithm*

12. *Implementation of MergeSort*

13. *Implementation of Quick Sort*

14. *Implementation of Data Searching using divide and conquer technique*


**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**COURSE STRUCTURE AND SYLLABUS**

**For**

**B. Tech COMPUTER SCIENCE &ENGINEERING**

*(Applicable for batches admitted from 2019-2020)*

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY: KAKINADA**

**KAKINADA - 533 003, Andhra Pradesh, India**

II Year – II Semester		<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>
<b>PROBABILITY AND STATISTICS</b>					

**Course Objectives:**

- To familiarize the students with the foundations of probability and statistical methods
- To impart probability concepts and statistical methods in various applications Engineering

**Course Outcomes:**

Upon successful completion of this course, the student should be able to

- Classify the concepts of data science and its importance (L4) or (L2)
- Interpret the association of characteristics and through correlation and regression tools (L4)
- Make use of the concepts of probability and their applications (L3)
- Apply discrete and continuous probability distributions (L3)
- Design the components of a classical hypothesis test (L6)
- Infer the statistical inferential methods based on small and large sampling tests (L4)

**UNIT I**

Descriptive statistics and methods for data science: Data science – Statistics Introduction – Population vs Sample – Collection of data – primary and secondary data – Type of variable: dependent and independent Categorical and Continuous variables – Data visualization – Measures of Central tendency – Measures of Variability (spread or variance) – Skewness Kurtosis.

**UNIT II**

Correlation and Curve fitting: Correlation – correlation coefficient – rank correlation – regression coefficients and properties – regression lines – Method of least squares – Straight line – parabola – Exponential – Power curves.

**UNIT III**

Probability and Distributions: Probability – Conditional probability and Baye's theorem – Random variables – Discrete and Continuous random variables – Distribution function – Mathematical Expectation and Variance – Binomial, Poisson, Uniform and Normal distributions.

**UNIT IV**

Sampling Theory: Introduction – Population and samples – Sampling distribution of Means and Variance (definition only) – Central limit theorem (without proof) – Introduction to t,  $\chi^2$  and F-distributions – Point and Interval estimations – Maximum error of estimate.

**UNIT V**

Tests of Hypothesis: Introduction – Hypothesis – Null and Alternative Hypothesis – Type I and Type II errors – Level of significance – One tail and two-tail tests – Tests concerning one mean and two means (Large and Small samples) – Tests on proportions.

**Text Books:**

- 1) Miller and Freund's, Probability and Statistics for Engineers, 7/e, Pearson, 2008.
- 2) S. C. Gupta and V.K. Kapoor, Fundamentals of Mathematical Statistics, 11/e, Sultan Chand & Sons Publications, 2012.

**Reference Books:**

- 1) Shron L. Myers, Keying Ye, Ronald E Walpole, Probability and Statistics Engineers and the Scientists, 8<sup>th</sup> Edition, Pearson 2007.
- 2) Jay I. Devore, Probability and Statistics for Engineering and the Sciences, 8<sup>th</sup> Edition, Cengage.
- 3) Sheldon M. Ross, Introduction to probability and statistics Engineers and the Scientists, 4<sup>th</sup> Edition, Academic Foundation, 2011.
- 4) Johannes Ledolter and Robert V. Hogg, Applied statistics for Engineers and Physical Scientists, 3<sup>rd</sup> Edition, Pearson, 2010.

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

<b>II Year – I Semester</b>		<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>
<b>UNIX and SHELL PROGRAMMING</b>					

**OBJECTIVES:**

- Written technical communication and effective use of concepts and terminology.
- Facility with UNIX command syntax and semantics.
- Ability to read and understand specifications, scripts and programs.
- Individual capability in problem solving using the tools presented within the class. Students will demonstrate a mastery of the course materials and concepts within in class discussions.

**UNIT-I** Introduction to unix-Brief History-What is Unix-Unix Components-Using Unix-Commands in Unix-Some Basic Commands-Command Substitution-Giving Multiple Commands.

**UNIT-II** The File system –The Basics of Files-What’s in a File-Directories and File Names-Permissions-I Nodes-The Directory Hierarchy, File Attributes and Permissions-The File Command knowing the File Type-The Chmod Command Changing File Permissions-The Chown Command Changing the Owner of a File-The Chgrp Command Changing the Group of a File.

**UNIT-III** Using the Shell-Command Line Structure-Met characters-Creating New Commands-Command Arguments and Parameters-Program Output as Arguments-Shell Variables- -More on I/O Redirection-Looping in Shell Programs.

**UNIT-IV** Filters-The Grep Family-Other Filters-The Stream Editor Sed-The AWK Pattern Scanning and processing Language-Good Files and Good Filters.

**UNIT-V** Shell Programming-Shell Variables-The Export Command-The Profile File a Script Run During Starting-The First Shell Script-The read Command-Positional parameters-The \$? Variable knowing the exit Status-More about the Set Command-The Exit Command-Branching Control Structures-Loop Control Structures-The Continue and Break Statement-The Expr Command: Performing Integer Arithmetic-Real Arithmetic in Shell Programs-The here Document(<<)-The Sleep Command-Debugging Scripts-The Script Command-The Eval Command-The Exec Command-The Process-The Meaning-Parent and Child Processes-Types of Processes-More about Foreground and Background processes-Internal and External Commands-Process Creation-The Trap Command-The Stty Command-The Kill Command-Job Control.



**OUTCOMES:** • Documentation will demonstrate good organization and readability. • File processing projects will require data organization, problem solving and research. • Scripts and programs will demonstrate simple effective user interfaces. • Scripts and programs will demonstrate effective use of structured programming. • Scripts and programs will be accompanied by printed output demonstrating completion of a test plan. • Testing will demonstrate both black and glass box testing strategies. • Project work will involve group participation.

**TEXT BOOKS:**

1. The Unix programming Environment by Brain W. Kernighan & Rob Pike, Pearson.
2. Introduction to Unix Shell Programming by M.G.Venkateshmurthy, Pearson.

**REFERENCE BOOKS:**

1. Unix and shell programming by B.M. Harwani, OXFORD university press.

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

<b>II Year – I Semester</b>		<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>
<b>JAVA PROGRAMMING</b>					

**Course Objectives:**

The learning objectives of this course are:

- To identify Java language components and how they work together in applications
- To learn the fundamentals of object-oriented programming in Java, including defining classes, invoking methods, using class libraries.
- To learn how to extend Java classes with inheritance and dynamic binding and how to use exception handling in Java applications
- To understand how to design applications with threads in Java
- To understand how to use Java APIs for program development

**Course Outcomes:**

By the end of the course, the student will be

- Able to realize the concept of Object Oriented Programming & Java Programming Constructs
- Able to describe the basic concepts of Java such as operators, classes, objects, inheritance, packages, Enumeration and various keywords
- Apply the concept of exception handling and Input/ Output operations
- Able to design the applications of Java & Java applet
- Able to Analyze & Design the concept of Event Handling and Abstract Window Toolkit

**UNIT I**

Program Structure in Java: Introduction, Writing Simple Java Programs, Elements or Tokens in Java Programs, Java Statements, Command Line Arguments, User Input to Programs, Escape Sequences Comments, Programming Style.

Data Types, Variables, and Operators :Introduction, Data Types in Java, Declaration of Variables, Data Types, Type Casting, Scope of Variable Identifier, Literal Constants, Symbolic Constants, Formatted Output with printf() Method, Static Variables and Methods, Attribute Final, Introduction to Operators, Precedence and Associativity of Operators, Assignment Operator ( = ), Basic Arithmetic Operators, Increment (++) and Decrement (- -) Operators, Ternary Operator, Relational Operators, Boolean Logical Operators, Bitwise Logical Operators.

Control Statements: Introduction, if Expression, Nested if Expressions, if-else Expressions, Ternary Operator?., Switch Statement, Iteration Statements, while Expression, do-while Loop, for Loop, Nested for Loop, For-Each for Loop, Break Statement, Continue Statement.

**UNIT II**

Classes and Objects: Introduction, Class Declaration and Modifiers, Class Members, Declaration of Class Objects, Assigning One Object to Another, Access Control for Class Members, Accessing Private Members of Class, Constructor Methods for Class, Overloaded Constructor Methods, Nested Classes, Final Class and Methods, Passing Arguments by Value and by

Reference, Keyword this.

Methods: Introduction, Defining Methods, Overloaded Methods, Overloaded Constructor Methods, Class Objects as Parameters in Methods, Access Control, Recursive Methods, Nesting of Methods, Overriding Methods, Attributes Final and Static.

### **UNIT III**

Arrays: Introduction, Declaration and Initialization of Arrays, Storage of Array in Computer Memory, Accessing Elements of Arrays, Operations on Array Elements, Assigning Array to Another Array, Dynamic Change of Array Size, Sorting of Arrays, Search for Values in Arrays, Class Arrays, Two-dimensional Arrays, Arrays of Varying Lengths, Three-dimensional Arrays, Arrays as Vectors.

Inheritance: Introduction, Process of Inheritance, Types of Inheritances, Universal Super Class-Object Class, Inhibiting Inheritance of Class Using Final, Access Control and Inheritance, Multilevel Inheritance, Application of Keyword Super, Constructor Method and Inheritance, Method Overriding, Dynamic Method Dispatch, Abstract Classes, Interfaces and Inheritance.

Interfaces: Introduction, Declaration of Interface, Implementation of Interface, Multiple Interfaces, Nested Interfaces, Inheritance of Interfaces, Default Methods in Interfaces, Static Methods in Interface, Functional Interfaces, Annotations.

### **UNIT IV**

Packages and Java Library: Introduction, Defining Package, Importing Packages and Classes into Programs, Path and Class Path, Access Control, Packages in Java SE, Java.lang Package and its Classes, Class Object, Enumeration, class Math, Wrapper Classes, Auto-boxing and Auto-unboxing, Java util Classes and Interfaces, Formatter Class, Random Class, Time Package, Class Instant (java.time.Instant), Formatting for Date/Time in Java, Temporal Adjusters Class, Temporal Adjusters Class.

Exception Handling: Introduction, Hierarchy of Standard Exception Classes, Keywords throws and throw, try, catch, and finally Blocks, Multiple Catch Clauses, Class Throwable, Unchecked Exceptions, Checked Exceptions, try-with-resources, Catching Subclass Exception, Custom Exceptions, Nested try and catch Blocks, Rethrowing Exception, Throws Clause.

### **UNIT V**

String Handling in Java: Introduction, Interface Char Sequence, Class String, Methods for Extracting Characters from Strings, Methods for Comparison of Strings, Methods for Modifying Strings, Methods for Searching Strings, Data Conversion and Miscellaneous Methods, Class String Buffer, Class String Builder.

Multithreaded Programming: Introduction, Need for Multiple Threads Multithreaded Programming for Multi-core Processor, Thread Class, Main Thread- Creation of New Threads, Thread States, Thread Priority-Synchronization, Deadlock and Race Situations, Inter-thread Communication - Suspending, Resuming, and Stopping of Threads.

Java Database Connectivity: Introduction, JDBC Architecture, Installing MySQL and MySQL Connector/J, JDBC Environment Setup, Establishing JDBC Database Connections, ResultSet Interface, Creating JDBC Application, JDBC Batch Processing, JDBC Transaction Management

### **Text Books:**

- 1) JAVA one step ahead, Anitha Seth, B.L.Juneja, Oxford.

- 2) The complete Reference Java, 8th edition, Herbert Schildt, TMH.

**References Books:**

- 1) Introduction to java programming, 7th edition by Y Daniel Liang, Pearson
- 2) Murach's Java Programming, Joel Murach

**esources:**

- 1) <https://nptel.ac.in/courses/106/105/106105191/>
- 2) [https://www.w3schools.com/java/java\\_data\\_types.asp](https://www.w3schools.com/java/java_data_types.asp)

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

<b>II Year – I Semester</b>		<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>
<b>ADVANCED DATA STRUCTURES &amp; ALGORITHMS</b>					

**Course Objectives:**

The objective of the course is to

- Introduce the fundamental concept of data structures and abstract data types
- Emphasize the importance of data structures in developing and implementing efficient algorithms
- Describe how arrays, records, linked structures, stacks, queues, trees, and graphs are represented in memory and used by algorithms

**Course Outcomes:**

After completing this course a student will be able to:

- Summarize the properties, interfaces, and behaviors of basic abstract data types
- Discuss the computational efficiency of the principal algorithms for sorting & searching
- Use arrays, records, linked structures, stacks, queues, trees, and Graphs in writing programs
- Demonstrate different methods for traversing trees

**UNIT I**

Data Structures - Definition, Classification of Data Structures, Operations on Data Structures, Abstract Data Type (ADT), Preliminaries of algorithms. Time and Space complexity.

Searching - Linear search, Binary search, Fibonacci search.

Sorting- Insertion sort, Selection sort, Exchange (Bubble sort, quick sort), distribution (radix sort), merging (Merge sort) algorithms.

**UNIT II**

Linked List: Introduction, Single linked list, Representation of Linked list in memory, Operations on Single Linked list-Insertion, Deletion, Search and Traversal ,Reversing Single Linked list, Applications on Single Linked list- Polynomial Expression Representation ,Addition and Multiplication, Sparse Matrix Representation using Linked List, Advantages and Disadvantages of Single Linked list, Double Linked list-Insertion, Deletion, Circular Linked list-Insertion, Deletion.

**UNIT III**

Queues: Introduction to Queues, Representation of Queues-using Arrays and using Linked list, Implementation of Queues-using Arrays and using Linked list, Application of Queues-Circular Queues, Deques, Priority Queues, Multiple Queues.

Stacks: Introduction to Stacks, Array Representation of Stacks, Operations on Stacks, Linked list Representation of Stacks, Operations on Linked Stack, Applications-Reversing list, Factorial Calculation, Infix to Postfix Conversion, Evaluating Postfix Expressions.

**UNIT IV**

Trees: Basic Terminology in Trees, Binary Trees-Properties, Representation of Binary Trees using Arrays and Linked lists. Binary Search Trees- Basic Concepts, BST Operations: Insertion, Deletion, Tree Traversals, Applications-Expression Trees, Heap Sort, Balanced Binary Trees-AVL Trees, Insertion, Deletion and Rotations.

**UNIT V**

Graphs: Basic Concepts, Representations of Graphs-Adjacency Matrix and using Linked list, Graph Traversals (BFT & DFT), Applications- Minimum Spanning Tree Using Prims & Kruskals Algorithm, Dijkstra's shortest path, Transitive closure, Warshall's Algorithm.

**Text Books:**

- 1) Data Structures Using C. 2<sup>nd</sup> Edition. Reema Thareja, Oxford.
- 2) Data Structures and algorithm analysis in C, 2<sup>nd</sup>ed, Mark Allen Weiss.

**Reference Books:**

- 1) Fundamentals of Data Structures in C, 2nd Edition, Horowitz, Sahni, Universities Press.
- 2) Data Structures: A PseudoCode Approach, 2/e, Richard F. Gilberg, Behrouz A. Forouzon, Cengage.
- 3) Data Structures with C, Seymour Lipschutz TMH

**Resources:**

- 1) <http://algs4.cs.princeton.edu/home/>
- 2) [https://faculty.washington.edu/jstraub/dsa/Master\\_2\\_7a.pdf](https://faculty.washington.edu/jstraub/dsa/Master_2_7a.pdf)

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

<b>II Year – I Semester</b>		<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>
<b>DIGITAL LOGIC DESIGN</b>					

**Course objectives:**

- To study the basic philosophy underlying the various number systems, negative number representation, binary arithmetic, theory of Boolean algebra and map method for minimization of switching functions.
- To introduce the basic tools for design of combinational and sequential digital logic.
- To learn simple digital circuits in preparation for computer engineering.

**Course outcomes:**

A student who successfully fulfills the course requirements will have demonstrated:

- An ability to define different number systems, binary addition and subtraction, 2's complement representation and operations with this representation.
- An ability to understand the different switching algebra theorems and apply them for logic functions.
- An ability to define the Karnaugh map for a few variables and perform an algorithmic reduction of logic functions.
- Students will be able to design various logic gates starting from simple ordinary gates to complex programmable logic devices & arrays.
- Students will be able to design various sequential circuits starting from flip-flop to registers and counters.

**UNIT I: Digital Systems and Binary Numbers**

Digital Systems, Binary Numbers, Octal and Hexadecimal Numbers, Complements of Numbers, Signed Binary Numbers, Arithmetic addition and subtraction, 4-bit codes: BCD, EXCESS 3, alphanumeric codes, 9's complement, 2421, etc..

**UNIT II: Concept of Boolean algebra**

Basic Theorems and Properties of Boolean algebra, Boolean Functions, Canonical and Standard Forms, Minterms and Maxterms.

## Gate level Minimization

Map Method, Three-Variable K-Map, Four Variable K-Maps. Products of Sum Simplification, Sum of Products Simplification, Don't – Care Conditions, NAND and NOR Implementation, ExclusiveOR Function.

**UNIT III: Combinational Logic**

Introduction, Analysis Procedure, Binary Adder–Subtractor, Binary Multiplier, Decoders, Encoders, Multiplexers, Demultiplexers, Priority Encoder, Code Converters, Magnitude Comparator, HDL Models of Combinational Circuits.

Realization of Switching Functions Using PROM, PAL and PLA.

**UNIT IV: Synchronous Sequential Logic**

Introduction to Sequential Circuits, Storage Elements: Latches, Flip-Flops, RS- Latch Using NAND and NOR Gates, Truth Tables. RS, JK, T and D Flip Flops, Truth and Excitation Tables, Conversion of Flip Flops.

**UNIT V: Registers and Counters**

Registers, Shift Registers, Ripple Counters, Synchronous Counters, Ring Counter, Johnson Counter.

**Text Books:**

- 1) Digital Design, 5/e, M.Morris Mano, Michael D Ciletti, PEA.
- 2) Fundamentals of Logic Design, 5/e, Roth, Cengage.

**Reference Books:**

- 1) Digital Logic and Computer Design, M.Morris Mano, PEA.
- 2) Digital Logic Design, Leach, Malvino, Saha, TMH.
- 3) Modern Digital Electronics, R.P. Jain, TMH.



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

<b>II Year – I Semester</b>		<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>0</b>	<b>0</b>	<b>3</b>	<b>1.5</b>
<b>ADVANCED DATASTRUCTURES LAB</b>					

**OBJECTIVES:**

- To understand heap and various tree structures like AVL, Red-black, B and Segment trees
- To understand the problems such as line segment intersection, convex shell and Voronoi diagram

**OUTCOMES:**

- Implement heap and various tree structure like AVL, Red-black, B and Segment trees
  - Solve the problems such as line segment intersection, convex shell and Voronoi diagram
1. To perform various operations i.e., insertions and deletions on AVL trees.
  2. To implement operations on binary heap.
    - a. Vertex insertion
    - b. Vertex deletion
    - c. Finding vertex
    - d. Edge addition and deletion
  3. To implement Prim's algorithm to generate a min-cost spanning tree.
  4. To implement Krushkal's algorithm to generate a min-cost spanning tree.
  5. To implement Dijkstra's algorithm to find shortest path in the graph.
  6. To implementation of Static Hashing (Use Linear probing for collision resolution)
  7. To implement of Huffmann coding.
  8. To implement of B-tree.

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

<b>II Year – I Semester</b>		<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>0</b>	<b>0</b>	<b>3</b>	<b>1.5</b>
<b>UNIX AND SHELL PROGRAMMING LAB</b>					

1. Demonstrate how to use the following Bourne Shell commands: cat, grep, ls, more, ps, chmod, finger, ftp, etc.
2. Use the following Bourne Shell constructs: test, if then, if then else, if then elif, for, while, until, and case.
3. Learn tracing mechanisms (for debugging), user variables, BourneShell variables, read-only variables, positional parameters, reading input to a BourneShell script, command substitution, comments, and exporting variables. In addition, test on numeric values, test on file type, and test on character strings are covered.
4. Copy, move, and delete files and directories
5. Write moderately complex Shell scripts.
6. Make a Shell script executable.
7. Create a ".profile" script to customize the user environment.
8. Use advanced features of File Transfer Protocol (FTP)
9. Compile source code into object and executable modules.
10. Execute programs written in C under UNIX environment

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

<b>II Year – I Semester</b>		<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>0</b>	<b>0</b>	<b>3</b>	<b>1.5</b>
<b>JAVA PROGRAMMING LAB</b>					

**Course Objectives:**

The aim of this lab is to

- Practice programming in the Java
- Gain knowledge of object-oriented paradigm in the Java programming language
- Learn use of Java in a variety of technologies and on different platforms

**Course Outcomes:**

By the end of the course student will be able to write java program for

- Evaluate default value of all primitive data type, Operations, Expressions, Control-flow, Strings
- Determine Class, Objects, Methods, Inheritance, Exception, Runtime Polymorphism, User defined Exception handling mechanism
- Illustrating simple inheritance, multi-level inheritance, Exception handling mechanism
- Construct Threads, Event Handling, implement packages, developing applets

**Exercise - 1 (Basics)**

- Write a JAVA program to display default value of all primitive data type of JAVA
- Write a java program that display the roots of a quadratic equation  $ax^2+bx=0$ . Calculate the discriminate D and basing on value of D, describe the nature of root.
- Five Bikers Compete in a race such that they drive at a constant speed which may or may not be the same as the other. To qualify the race, the speed of a racer must be more than the average speed of all 5 racers. Take as input the speed of each racer and print back the speed of qualifying racers.

**Exercise - 2 (Operations, Expressions, Control-flow, Strings)**

- Write a JAVA program to search for an element in a given list of elements using binary search mechanism.
- Write a JAVA program to sort for an element in a given list of elements using bubble sort
- Write a JAVA program to sort for an element in a given list of elements using merge sort.
- Write a JAVA program using StringBuffer to delete, remove character.

**Exercise - 3 (Class, Objects)**

- Write a JAVA program to implement class mechanism. Create a class, methods and invoke them inside main method.
- Write a JAVA program to implement constructor.

**Exercise - 4 (Methods)**

- Write a JAVA program to implement constructor overloading.
- Write a JAVA program implement method overloading.

**Exercise - 5 (Inheritance)**

- Write a JAVA program to implement Single Inheritance
- Write a JAVA program to implement multi level Inheritance

c) Write a java program for abstract class to find areas of different shapes

**Exercise - 6 (Inheritance - Continued)**

a) Write a JAVA program give example for “super” keyword.

b) Write a JAVA program to implement Interface. What kind of Inheritance can be achieved?

**Exercise - 7 (Exception)**

- a) Write a JAVA program that describes exception handling mechanism
- b) Write a JAVA program Illustrating Multiple catch clauses

**Exercise – 8 (Runtime Polymorphism)**

- a) Write a JAVA program that implements Runtime polymorphism
- b) Write a Case study on run time polymorphism, inheritance that implements in above problem

**Exercise – 9 (User defined Exception)**

- a) Write a JAVA program for creation of Illustrating throw
- b) Write a JAVA program for creation of Illustrating finally
- c) Write a JAVA program for creation of Java Built-in Exceptions
- d) Write a JAVA program for creation of User Defined Exception

**Exercise – 10 (Threads)**

- a) Write a JAVA program that creates threads by extending Thread class .First thread display “Good Morning “every 1 sec, the second thread displays “Hello “every 2 seconds and the third display “Welcome” every 3 seconds ,(Repeat the same by implementing Runnable)
- b) Write a program illustrating **isAlive** and **join ()**
- c) Write a Program illustrating Daemon Threads.

**Exercise - 11 (Threads continuity)**

- a) Write a JAVA program Producer Consumer Problem
- b) Write a case study on thread Synchronization after solving the above producer consumer problem

**Exercise – 12 (Packages)**

- a) Write a JAVA program illustrate class path
- b) Write a case study on including in class path in your os environment of your package.
- c) Write a JAVA program that import and use the defined your package in the previous Problem

**Exercise - 13 (Applet)**

- a) Write a JAVA program to paint like paint brush in applet.
- b) Write a JAVA program to display analog clock using Applet.
- c) Write a JAVA program to create different shapes and fill colors using Applet.

**Exercise - 14 (Event Handling)**

- a) Write a JAVA program that display the x and y position of the cursor movement using Mouse.
- b) Write a JAVA program that identifies key-up key-down event user entering text in a Applet.

## DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

<b>II Year – I Semester</b>		<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>0</b>
<b>ESSENCE OF INDIAN TRADITIONAL KNOWLEDGE</b>					

### Course Objectives:

To facilitate the students with the concepts of Indian traditional knowledge and to make them understand the Importance of roots of knowledge system

- The course aim of the importing basic principle of third process reasoning and inference sustainability is at the course of Indian traditional knowledge system
- To understand the legal framework and traditional knowledge and biological diversity act 2002 and geographical indication act 2003
- The courses focus on traditional knowledge and intellectual property mechanism of traditional knowledge and protection
- To know the student traditional knowledge in different sector

### Course Outcomes:

After completion of the course, students will be able to:

- Understand the concept of Traditional knowledge and its importance
- Know the need and importance of protecting traditional knowledge
- Know the various enactments related to the protection of traditional knowledge
- Understand the concepts of Intellectual property to protect the traditional knowledge

### UNIT I

Introduction to traditional knowledge: Define traditional knowledge, nature and characteristics, scope and importance, kinds of traditional knowledge, the physical and social contexts in which traditional knowledge develop, the historical impact of social change on traditional knowledge systems. Indigenous Knowledge (IK), characteristics, traditional knowledge vis-à-vis indigenous knowledge, traditional knowledge Vs western knowledge traditional knowledge vis-à-vis formal knowledge

Learning Outcomes:

At the end of the unit, the student will able to:

- Understand the traditional knowledge.
- Contrast and compare characteristics importance kinds of traditional knowledge.
- Analyze physical and social contexts of traditional knowledge.
- Evaluate social change on traditional knowledge.

### UNIT II

Protection of traditional knowledge: the need for protecting traditional knowledge Significance of TK Protection, value of TK in global economy, Role of Government to harness TK.

Learning Outcomes:

At the end of the unit, the student will able to:

- Know the need of protecting traditional knowledge.
- Apply significance of tk protection.
- Analyze the value of tk in global economy.
- Evaluate role of government

**UNIT III**

Legal framework and TK: A: The Scheduled Tribes and Other Traditional Forest Dwellers (Recognition of Forest Rights) Act, 2006, Plant Varieties Protection and Farmers Rights Act, 2001 (PPVFR Act); B: The Biological Diversity Act 2002 and Rules 2004, the protection of traditional knowledge bill, 2016. Geographical indications act 2003.

Learning Outcomes:

At the end of the unit the student will be able to:

- Understand legal framework of TK.
- Contrast and compare the ST and other traditional forest dwellers
- Analyze plant variety protections
- Evaluate farmers right act

**UNIT IV**

Traditional knowledge and intellectual property: Systems of traditional knowledge protection, Legal concepts for the protection of traditional knowledge, Certain non IPR mechanisms of traditional knowledge protection, Patents and traditional knowledge, Strategies to increase protection of traditional knowledge, global legal FORA for increasing protection of Indian Traditional Knowledge.

Learning Outcomes:

At the end of the unit, the student will be able to:

- Understand TK and IPR
- Apply systems of TK protection.
- Analyze legal concepts for the protection of TK.
- Evaluate strategies to increase the protection of TK.

**UNIT V**

Traditional knowledge in different sectors: Traditional knowledge and engineering, Traditional medicine system, TK and biotechnology, TK in agriculture, Traditional societies depend on it for their food and healthcare needs, Importance of conservation and sustainable development of environment, Management of biodiversity, Food security of the country and protection of TK.

Learning Outcomes:

At the end of the unit, the student will be able to:

- Know TK in different sectors.
- Apply TK in engineering.
- Analyze TK in various sectors.
- Evaluate food security and protection of TK in the country.

**Reference Books:**

- 1) Traditional Knowledge System in India, by Amit Jha, 2009.
- 2) Traditional Knowledge System and Technology in India by Basanta Kumar Mohanta and Vipin Kumar Singh, PratibhaPrakashan 2012.
- 3) Traditional Knowledge System in India by Amit Jha Atlantic publishers, 2002
- 4) "Knowledge Traditions and Practices of India" Kapil Kapoor, Michel Danino

**Resources:**

- 1) <https://www.youtube.com/watch?v=LZP1StpYEPM>
- 2) <http://nptel.ac.in/courses/121106003/>

## DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

<b>II Year – II Semester</b>		<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>2</b>	<b>1</b>	<b>0</b>	<b>3</b>
<b>JAVA PROGRAMMING</b>					

### Course Objectives:

The learning objectives of this course are:

- To identify Java language components and how they work together in applications
- To learn the fundamentals of object-oriented programming in Java, including defining classes, invoking methods, using class libraries.
- To learn how to extend Java classes with inheritance and dynamic binding and how to use exception handling in Java applications
- To understand how to design applications with threads in Java
- To understand how to use Java APIs for program development

### Course Outcomes:

By the end of the course, the student will be

- Able to realize the concept of Object Oriented Programming & Java Programming Constructs
- Able to describe the basic concepts of Java such as operators, classes, objects, inheritance, packages, Enumeration and various keywords
- Apply the concept of exception handling and Input/ Output operations
- Able to design the applications of Java & Java applet
- Able to Analyze & Design the concept of Event Handling and Abstract Window Toolkit

### UNIT I

Program Structure in Java: Introduction, Writing Simple Java Programs, Elements or Tokens in Java Programs, Java Statements, Command Line Arguments, User Input to Programs, Escape Sequences Comments, Programming Style.

Data Types, Variables, and Operators :Introduction, Data Types in Java, Declaration of Variables, Data Types, Type Casting, Scope of Variable Identifier, Literal Constants, Symbolic Constants, Formatted Output with printf() Method, Static Variables and Methods, Attribute Final, Introduction to Operators, Precedence and Associativity of Operators, Assignment Operator ( = ), Basic Arithmetic Operators, Increment (++) and Decrement (- -) Operators, Ternary Operator, Relational Operators, Boolean Logical Operators, Bitwise Logical Operators.

Control Statements: Introduction, if Expression, Nested if Expressions, if–else Expressions, Ternary Operator?;, Switch Statement, Iteration Statements, while Expression, do–while Loop, for Loop, Nested for Loop, For–Each for Loop, Break Statement, Continue Statement.

### UNIT II

Classes and Objects: Introduction, Class Declaration and Modifiers, Class Members, Declaration of Class Objects, Assigning One Object to Another, Access Control for Class Members, Accessing Private Members of Class, Constructor Methods for Class, Overloaded Constructor Methods, Nested Classes, Final Class and Methods, Passing Arguments by Value and by



## **DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

Reference, Keyword this.

Methods: Introduction, Defining Methods, Overloaded Methods, Overloaded Constructor Methods, Class Objects as Parameters in Methods, Access Control, Recursive Methods, Nesting of Methods, Overriding Methods, Attributes Final and Static.

### **UNIT III**

Arrays: Introduction, Declaration and Initialization of Arrays, Storage of Array in Computer Memory, Accessing Elements of Arrays, Operations on Array Elements, Assigning Array to Another Array, Dynamic Change of Array Size, Sorting of Arrays, Search for Values in Arrays, Class Arrays, Two-dimensional Arrays, Arrays of Varying Lengths, Three-dimensional Arrays, Arrays as Vectors.

Inheritance: Introduction, Process of Inheritance, Types of Inheritances, Universal Super Class-Object Class, Inhibiting Inheritance of Class Using Final, Access Control and Inheritance, Multilevel Inheritance, Application of Keyword Super, Constructor Method and Inheritance, Method Overriding, Dynamic Method Dispatch, Abstract Classes, Interfaces and Inheritance.

Interfaces: Introduction, Declaration of Interface, Implementation of Interface, Multiple Interfaces, Nested Interfaces, Inheritance of Interfaces, Default Methods in Interfaces, Static Methods in Interface, Functional Interfaces, Annotations.

### **UNIT IV**

Packages and Java Library: Introduction, Defining Package, Importing Packages and Classes into Programs, Path and Class Path, Access Control, Packages in Java SE, Java.lang Package and its Classes, Class Object, Enumeration, class Math, Wrapper Classes, Auto-boxing and Auto-unboxing, Java util Classes and Interfaces, Formatter Class, Random Class, Time Package, Class Instant (java.time.Instant), Formatting for Date/Time in Java, Temporal Adjusters Class, Temporal Adjusters Class.

Exception Handling: Introduction, Hierarchy of Standard Exception Classes, Keywords throws and throw, try, catch, and finally Blocks, Multiple Catch Clauses, Class Throwable, Unchecked Exceptions, Checked Exceptions, try-with-resources, Catching Subclass Exception, Custom Exceptions, Nested try and catch Blocks, Rethrowing Exception, Throws Clause.

### **UNIT V**

String Handling in Java: Introduction, Interface Char Sequence, Class String, Methods for Extracting Characters from Strings, Methods for Comparison of Strings, Methods for Modifying Strings, Methods for Searching Strings, Data Conversion and Miscellaneous Methods, Class String Buffer, Class String Builder.

Multithreaded Programming: Introduction, Need for Multiple Threads Multithreaded Programming for Multi-core Processor, Thread Class, Main Thread- Creation of New Threads, Thread States, Thread Priority-Synchronization, Deadlock and Race Situations, Inter-thread Communication - Suspending, Resuming, and Stopping of Threads.

Java Database Connectivity: Introduction, JDBC Architecture, Installing MySQL and MySQL Connector/J, JDBC Environment Setup, Establishing JDBC Database Connections, ResultSet Interface, Creating JDBC Application, JDBC Batch Processing, JDBC Transaction Management

## DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

### **Text Books:**

- 3) JAVA one step ahead, Anitha Seth, B.L.Juneja, Oxford.
- 4) The complete Reference Java, 8th edition, Herbert Schildt, TMH.

### **References Books:**

- 3) Introduction to java programming, 7th edition by Y Daniel Liang, Pearson
- 4) Murach's Java Programming, Joel Murach

### **esources:**

- 1) <https://nptel.ac.in/courses/106/105/106105191/>
- 2) [https://www.w3schools.com/java/java\\_data\\_types.asp](https://www.w3schools.com/java/java_data_types.asp)





## DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

<b>II Year – II Semester</b>		<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>0</b>	<b>0</b>	<b>3</b>	<b>1.5</b>
<b>JAVA PROGRAMMING LAB</b>					

### Course Objectives:

The aim of this lab is to

- Practice programming in the Java
- Gain knowledge of object-oriented paradigm in the Java programming language
- Learn use of Java in a variety of technologies and on different platforms

### Course Outcomes:

By the end of the course student will be able to write java program for

- Evaluate default value of all primitive data type, Operations, Expressions, Control-flow, Strings
- Determine Class, Objects, Methods, Inheritance, Exception, Runtime Polymorphism, User defined Exception handling mechanism
- Illustrating simple inheritance, multi-level inheritance, Exception handling mechanism
- Construct Threads, Event Handling, implement packages, developing applets

### Exercise - 1 (Basics)

- d) Write a JAVA program to display default value of all primitive data type of JAVA
- e) Write a java program that display the roots of a quadratic equation  $ax^2+bx=0$ . Calculate the discriminate D and basing on value of D, describe the nature of root.
- f) Five Bikers Compete in a race such that they drive at a constant speed which may or may not be the same as the other. To qualify the race, the speed of a racer must be more than the average speed of all 5 racers. Take as input the speed of each racer and print back the speed of qualifying racers.

### Exercise - 2 (Operations, Expressions, Control-flow, Strings)

- e) Write a JAVA program to search for an element in a given list of elements using binary search mechanism.
- f) Write a JAVA program to sort for an element in a given list of elements using bubble sort
- g) Write a JAVA program to sort for an element in a given list of elements using merge sort.
- h) Write a JAVA program using StringBuffer to delete, remove character.

### Exercise - 3 (Class, Objects)

- c) Write a JAVA program to implement class mechanism. Create a class, methods and invoke them inside main method.
- d) Write a JAVA program to implement constructor.

### Exercise - 4 (Methods)

- c) Write a JAVA program to implement constructor overloading.
- d) Write a JAVA program implement method overloading.

### Exercise - 5 (Inheritance)

- d) Write a JAVA program to implement Single Inheritance
- e) Write a JAVA program to implement multi level Inheritance
- f) Write a java program for abstract class to find areas of different shapes

### Exercise - 6 (Inheritance - Continued)

- c) Write a JAVA program give example for “super” keyword.
- d) Write a JAVA program to implement Interface. What kind of Inheritance can be achieved?

## DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

### Exercise - 7 (Exception)

- c) Write a JAVA program that describes exception handling mechanism
- d) Write a JAVA program Illustrating Multiple catch clauses

### Exercise – 8 (Runtime Polymorphism)

- c) Write a JAVA program that implements Runtime polymorphism
- d) Write a Case study on run time polymorphism, inheritance that implements in above problem

### Exercise – 9 (User defined Exception)

- e) Write a JAVA program for creation of Illustrating throw
- f) Write a JAVA program for creation of Illustrating finally
- g) Write a JAVA program for creation of Java Built-in Exceptions
- h) d)Write a JAVA program for creation of User Defined Exception

### Exercise – 10 (Threads)

- d) Write a JAVA program that creates threads by extending Thread class .First thread display “Good Morning “every 1 sec, the second thread displays “Hello “every 2 seconds and the third display “Welcome” every 3 seconds ,(Repeat the same by implementing Runnable)
- e) Write a program illustrating **isAlive** and **join ()**
- f) Write a Program illustrating Daemon Threads.

### Exercise - 11 (Threads continuity)

- c) Write a JAVA program Producer Consumer Problem
- d) Write a case study on thread Synchronization after solving the above producer consumer problem

### Exercise – 12 (Packages)

- d) Write a JAVA program illustrate class path
- e) Write a case study on including in class path in your os environment of your package.
- f) Write a JAVA program that import and use the defined your package in the previous Problem

### Exercise - 13 (Applet)

- d) Write a JAVA program to paint like paint brush in applet.
- e) Write a JAVA program to display analog clock using Applet.
- f) Write a JAVA program to create different shapes and fill colors using Applet.

### Exercise - 14 (Event Handling)

- c) Write a JAVA program that display the x and y position of the cursor movement using Mouse.
- d) Write a JAVA program that identifies key-up key-down event user entering text in a Applet.

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

<b>II Year – II Semester</b>		<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>0</b>	<b>0</b>	<b>2</b>	<b>1</b>
<b>UNIX OPERATING SYSTEM LAB</b>					

**Course Objectives:**

- To understand the design aspects of operating system
- To study the process management concepts & Techniques
- To study the storage management concepts
- To familiarize students with the Linux environment
- To learn the fundamentals of shell scripting/programming

**Course Outcomes:**

- To use Unix utilities and perform basic shell control of the utilities
  - To use the Unix file system and file access control
  - To use of an operating system to develop software
  - Students will be able to use Linux environment efficiently
  - Solve problems using bash for shell scripting
- 1) a) Study of Unix/Linux general purpose utility command list: man, who, cat, cd, cp, ps, ls, mv, rm, mkdir, rmdir, echo, more, date, time, kill, history, chmod, chown, finger, pwd, cal, logout, shutdown.
    - b) Study of vi editor
    - c) Study of Bash shell, Bourne shell and C shell in Unix/Linux operating system
    - d) Study of Unix/Linux file system (tree structure)
    - e) Study of .bashrc, /etc/bashrc and Environment variables.
  - 2) Write a C program that makes a copy of a file using standard I/O, and system calls
  - 3) Write a C program to emulate the UNIX ls -l command.
  - 4) Write a C program that illustrates how to execute two commands concurrently with a command pipe. Ex: - ls -l | sort
  - 5) Simulate the following CPU scheduling algorithms:
    - (a) Round Robin (b) SJF (c) FCFS (d) Priority
  - 6) Multiprogramming-Memory management-Implementation of fork (), wait (), exec() and exit (), System calls
  - 7) Simulate the following:
    - a) Multiprogramming with a fixed number of tasks (MFT)
    - b) Multiprogramming with a variable number of tasks (MVT)
  - 8) Simulate Bankers Algorithm for Dead Lock Avoidance
  - 9) Simulate Bankers Algorithm for Dead Lock Prevention.

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

- 10) Simulate the following page replacement algorithms:  
a) FIFO b) LRU c) LFU
- 11) Simulate the following File allocation strategies  
(a) Sequenced (b) Indexed (c) Linked
- 12) Write a C program that illustrates two processes communicating using shared memory
- 13) Write a C program to simulate producer and consumer problem using semaphores
- 14) Write C program to create a thread using pthreads library and let it run its function.
- 15) Write a C program to illustrate concurrent execution of threads using pthreads library.







**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**COURSE STRUCTURE AND SYLLABUS**

**For**

**B. Tech COMPUTER SCIENCE &ENGINEERING**

*(Applicable for batches admitted from 2019-2020)*

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY: KAKINADA**

**KAKINADA - 533 003, Andhra Pradesh, India**

## DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

<b>II Year – II Semester</b>		<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>1</b>	<b>0</b>	<b>4</b>
<b>PHP PROGRAMMING</b>					

### Course Objectives:

This course is designed to:

- To introduce the students to the topics and techniques of discrete methods and combinatorial reasoning
- To introduce a wide variety of applications. The algorithmic approach to the solution of problems is fundamental in discrete mathematics, and this approach reinforces the close ties between this discipline and the area of computer science

### Course Outcomes:

The theory should be taught and practical should be carried out in such a manner that students are able to acquire different learning outcomes in cognitive, psychomotor and affective domain to demonstrate following course outcomes.

- i. Create small programs using basic PHP concepts.
- ii. Apply In-Built and Create User defined functions in PHP programming.
- iii. Design and develop a Web site using form controls for presenting web based content.
- iv. Debug the Programmes by applying concepts and error handling techniques of PHP.
- v. Create dynamic Website/ Web based Applications, using PHP, MySQL database

## UNIT I

### Introduction to PHP:

Configuration of PHP, Apache Web Server, MySQL and Open Source, Relationship between Apache, MySQL and PHP(AMP Module), Installing PHP for (Windows, Wamp server, XAMP server), PHP Structure and Syntax, Creating PHP pages, Rules of PHP syntax, Integrating HTML with PHP Constants, Variables: Static and Global Variable, Conditional Structure and Looping, PHP operators, Arrays, constructs, User Defined function, argument function, variable function, Return function, default argument, variable length argument

## UNIT II

### Working with In Built Functions:

Variable Function:(gettype, settype, isset, strval, floatval, intval, print\_r) string function: (Chr, ord, strtolower, strtoupper, strlen, ltrim, rtrim, trim, substr, strcmp, strcasecmp, strpos, strrpos, strpos, str\_replace, strpos, echo, print), MATH functions: (Abs, ceil, floor, round, fmod, min, max, pow, sqrt, rand), Date function:(Date, getdate, setdate, checkdate, time, mktime) Array Function:(Count, list, in\_array, current, next, previous, end, each, sort, array\_merge, array\_reverse), File function: (Fopen, fread, fwrite, fclose)

## UNIT III

### Working with data and forms:

Reading data using Form Controls (Text Fields, Text Areas, CheckBoxes, Radio Buttons, List Boxes, Password Controls, Hidden Controls, Image Maps, File Uploads, Buttons), Submitting form values, using \$\_GET and \$\_POST Methods, \$\_REQUEST Accessing form inputs with Get/Post functions, Combining HTML and PHP codes together on single page, Redirecting the user

**UNIT IV****Session, Cookies and Error Handling:**

Setting a cookie with PHP, Deleting a cookie, Creating session cookie, Working with the query string, Creating query string, Session, Starting and Destroying session, Working with session variables, Passing session IDs, Error Types in PHP, Exception Handling in PHP.

**UNIT V****Database Connectivity using MYSQL: Concepts and Installation of MySQL**

MySQL structure and syntax, Types of MySQL tables and Storage engines, MySQL commands,

Integration of PHP with MySQL, Connection to the MySQL Database, Creating and Deleting MySQL database using PHP, Updating, Inserting, Deleting records in the MySQL database, Hosting Website (Using 'C' panel, Using Filezilla Software)

**Reference Books:**

1	Beginning PHP and MySQL, 4 <sup>th</sup> Edition	W. Jason Gilmore	Apress, 2010
2	PHP: The Complete Reference	Steven Holzner	McGraw-Hill, 2008
3	Learning PHP, MySQL, JavaScript, CSS & HTML5, Third Edition	Robin Nixon	O'reilly Media, 2014
4	Teach yourself PHP, MySQL and Apache All in One, 5 <sup>th</sup> Edition	Julie C. Meloni,	Pearson Education, 2012

**Resources:**

Software: WAMP server / XAMPP server, 'C' Panel, Text Editor

- i. <http://www.codecademy.com/tracks/web>,
- ii. <http://www.codecademy.com/tracks/php>
- iii. <http://www.w3schools.com/PHP>
- iv. <http://www.tutorialpoint.com>
- v. <http://www.homeandlearn.co.uk>

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

<b>II Year – II Semester</b>		<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>
<b>COMPUTER ORGANIZATION</b>					

**Course Objectives:**

The course objectives of Computer Organization are to discuss and make student familiar with the

- Principles and the Implementation of Computer Arithmetic
- Operation of CPUs including RTL, ALU, Instruction Cycle and Busses
- Fundamentals of different Instruction Set Architectures and their relationship to the CPU Design
- Memory System and I/O Organization
- Principles of Operation of Multiprocessor Systems and Pipelining

**Course Outcomes:**

By the end of the course, the student will

- Develop a detailed understanding of computer systems
- Cite different number systems, binary addition and subtraction, standard, floating-point, and micro operations
- Develop a detailed understanding of architecture and functionality of central processing unit
- Exemplify in a better way the I/O and memory organization
- Illustrate concepts of parallel processing, pipelining and inter processor communication

**UNIT I**

Basic Structure of Computers: Basic Organization of Computers, Historical Perspective, Bus Structures, Data Representation: Data types, Complements, Fixed Point Representation. Floating, Point Representation. Other Binary Codes, Error Detection Codes.

Computer Arithmetic: Addition and Subtraction, Multiplication Algorithms, Division Algorithms.

**UNIT II**

Register Transfer Language and Microoperations: Register Transfer language. Register Transfer Bus and Memory Transfers, Arithmetic Micro operations, Logic Micro Operations, Shift Micro Operations, Arithmetic Logic Shift Unit.

Basic Computer Organization and Design: Instruction Codes, Computer Register, Computer Instructions, Instruction Cycle, Memory – Reference Instructions. Input –Output and Interrupt, Complete Computer Description.

**UNIT III**

Central Processing Unit: General Register Organization, STACK Organization. Instruction Formats, Addressing Modes, Data Transfer and Manipulation, Program Control, Reduced Instruction Set Computer.

Microprogrammed Control: Control Memory, Address Sequencing, Micro Program example, Design of Control Unit.

**UNIT IV**

Memory Organization: Memory Hierarchy, Main Memory, Auxiliary Memory, Associative Memory, Cache Memory, Virtual Memory.

Input-Output Organization: Peripheral Devices, Input-Output Interface, Asynchronous data transfer, Modes of Transfer, Priority Interrupts, Direct Memory Access.

## UNIT V

Multi Processors: Introduction, Characteristics of Multiprocessors, Interconnection Structures, Inter Processor Arbitration.

Pipeline: Parallel Processing, Pipelining, Instruction Pipeline, RISC Pipeline, Array Processor.

### **Text Books:**

- 1) Computer System Architecture, M. Morris Mano, Third Edition, Pearson, 2008.
- 2) Computer Organization, Carl Hamacher, Zvonko Vranesic, Safwat Zaky, 5/e, McGraw Hill, 2002.

### **Reference Books:**

- 1) Computer Organization and Architecture, William Stallings, 6/e, Pearson, 2006.
- 2) Structured Computer Organization, Andrew S. Tanenbaum, 4/e, Pearson, 2005.
- 3) Fundamentals of Computer Organization and Design, Sivarama P. Dandamudi, Springer, 2006.

### **e- Resources:**

- 1) <https://nptel.ac.in/courses/106/105/106105163/>
- 2) <http://www.cuc.ucc.ie/CS1101/David%20Tarnoff.pdf>

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

<b>II Year – II Semester</b>		<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>
<b>ADVANCED JAVA PROGRAMMING</b>					

**Course Objectives:**

The learning objectives of this course are:

- To identify Java language components and how they work together in applications
- To learn the fundamentals of object-oriented programming in Java, including defining classes, invoking methods, using class libraries.
- To learn how to extend Java classes with inheritance and dynamic binding and how to use exception handling in Java applications
- To understand how to design applications with threads in Java
- To understand how to use Java APIs for program development

**Course Outcomes:**

By the end of the course, the student will be

To be familiarize with RMI and JSP

To understand the Java Servlets and Database connectivity.

To know more about the Enterprise Java Bean (EJB) Programming

**UNIT I**

**APPLET, AWT AND EVENT HANDLING** : Applet Basics – Applet architecture – HTML APPLET tag – Passing parameter to Applet.getDocumentBase() and getCodeBase() – AWT classes and Graphics – AWT Controls -Event Handling – Event Classes – Event Listener Interfaces – Layout Managers – Menus

**UNIT II**

**INTRODUCING SWING & JAVA BEANS**: Exploring Swing – JLabel and ImageIcon, JTextField – The Swing Buttons – JTabbedPane -JScrollPane, JList&JcomboBox – Trees &JTables – What Is a Java Bean? – Advantages of Java Beans – Introspection, Bound and Constrained Properties – Persistence & Customizers

**UNIT III**

**RMI & NETWORKING** : Remote Method Invocation – Settingup Remote Method Invocation – RMI with Applets -Networking Basics – The Networking Classes and Interfaces – InetAddress – Inet4Address and Inet6Address -TCP/IP Client sockets – URL – URL Connection – HttpURLConnection .

**UNIT IV**

**JDBC** : Presentation to JDBC CONNECTION settings – The Concept of JDBC – JDBC Driver Types – JDBC Packages – A Brief Overview of the JDBC Process – Database Connection – Associating the JDBC/ODBC Bridge with the Database – Statement Objects – Result Set.

**UNIT V**

**SERVLETS**: Background, The Life Cycle of a Servlet & The JSDK-A Simple Servlet – The Servlet API -RolePlay-Servlet Concept – The javax.servlet Package – Reading Servlet Parameters, The javax.servlet.http Package – Handling HTTP Request and Responses – Using Cookies – Session Tracking.



**Text Books:**

- 1) Naughton and H.Schildt, (2007), “Java 2-The complete reference”, Fifth Edition McGraw Hill.  
(UNIT I – V)

**References Books:**

- 1) Jim Keogh, (2002), “The Complete Reference J2EE”, Tata McGraw Hill Edition, New Delhi.
- 2) Marty Hall, Larry Brown, (2004), “Core Servlets and Java Server Pages”, 2<sup>nd</sup> Edition, Pearson Education.

<b>II Year – II Semester</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
	<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>
<b>R PROGRAMMING</b>				

**OBJECTIVE:**

Students will be able to Use R for statistical programming, computation, graphics, and modeling, Write functions and use R in an efficient way, Fit some basic types of statistical models Use R in their own research, Be able to expand their knowledge of R on their own.

**OUTCOMES:**

At the end of this course, students will be able to: List motivation for learning a programming language Access online resources for R and import new function packages into the R workspace Import, review, manipulate and summarize data-sets in R. Explore data-sets to create testable hypotheses and identify appropriate statistical tests Perform appropriate statistical tests using R Create and edit visualizations.

**UNIT-I:**

Introduction, How to run R, R Sessions and Functions, Basic Math, Variables, Data Types, Vectors, Conclusion, Advanced Data Structures, Data Frames, Lists, Matrices, Arrays, Classes.

**UNIT-II:**

R Programming Structures, Control Statements, Loops, - Looping Over Nonvector Sets,- If-Else, Arithmetic and Boolean Operators and values, Default Values for Argument, Return Values, Deciding Whether to explicitly call return- Returning Complex Objects, Functions are Objective, No Pointers in R, Recursion, A Quicksort Implementation-Extended Extended Example: A Binary Search Tree.

**UNIT-III:**

Doing Math and Simulation in R, Math Function, Extended Example Calculating Probability-Cumulative Sums and Products-Minima and Maxima- Calculus, Functions Fir Statistical Distribution, Sorting, Linear Algebra Operation on Vectors and Matrices, Extended Example: Vector cross Product- Extended Example: Finding Stationary Distribution of Markov Chains, Set Operation, Input /out put, Accessing the Keyboard and Monitor, Reading and writer Files,

**UNIT-IV:**

**Graphics, Creating Graphs, The Workhorse of R Base Graphics, the plot() Function – Customizing Graphs, Saving Graphs to Files.**

**UNIT-V:**

Probability Distributions, Normal Distribution- Binomial Distribution- Poisson Distributions Other Distribution, Basic Statistics, Correlation and Covariance, T-Tests,- ANOVA.

**UNIT-VI:**

Linear Models, Simple Linear Regression, -Multiple Regression Generalized Linear Models, Logistic Regression,  
- Poisson Regression- other Generalized Linear Models-Survival Analysis, Nonlinear Models, Splines- Decision- Random Forests,

**TEXT BOOKS:**

- 1) The Art of R Programming, A K Verma, Cengage Learning.
- 2) R for Everyone, Lander, Pearson
- 3) The Art of R Programming, Norman Matloff, No starch Press.

**REFERENCE BOOKS:**

- 1) R Cookbook, Paul Teetor, Oreilly.
- 2) R in Action, Rob Kabacoff, Manning

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

<b>II Year – I Semester</b>		<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>0</b>	<b>0</b>	<b>3</b>	<b>1.5</b>
<b>PHP PROGRAMMING LAB</b>					

**Course Objectives:**

The aim of Python Programming Lab is

- To acquire programming skills in core Python.
- To acquire Object Oriented Skills in Python
- To develop the skill of designing Graphical user Interfaces in Python
- To develop the ability to write database applications in Python

**Course Outcomes:**

By the end of this lab, the student is able to

- Write, Test and Debug Python Programs
- Use Conditionals and Loops for Python Programs
- Use functions and represent Compound data using Lists, Tuples and Dictionaries
- Use various applications using python

<b>S. No.</b>	<b>Practical Exercises</b>
1.	Write a PHP script to display Welcome message.
2.	Write a PHP script to demonstrate arithmetic operators, comparison operator, and logical operator.
3.	Write PHP Script to print Fibonacci series.
4.	Write PHP Script to generate result and display grade.
5.	Write PHP Script to find maximum number out of three given numbers.
6.	Write PHP Script for addition of two 2x2 matrices.
7.	Write PHP script to demonstrate Variable function.
8.	Write PHP script to obtain 5! Using function
9.	Write PHP script to demonstrate string function.
10.	Write PHP script to demonstrate Date functions.
11.	Write PHP script to demonstrate Math functions.
12.	Write PHP script to demonstrate Array functions.

S. No.	<p style="text-align: center;"><b>Practical Exercises</b> (Outcomes in Psychomotor Domain)</p>
13.	Write PHP script to demonstrate File functions.
14.	Create student registration form using text box, check box, radio button, select, submit button. And display user inserted value in new PHP page.
15.	Create Website Registration Form using text box, check box, radio button, select, submit button. And display user inserted value in new PHP page.
16.	Write two different PHP script to demonstrate passing variables through a URL.
17.	Write two different PHP script to demonstrate passing variables with sessions.
18.	Write PHP script to demonstrate passing variables with cookies.
19.	Write a program to keep track of how many times a visitor has loaded the page.
20.	Write an example of Error-handling using exceptions.
21.	Write a PHP script to connect MySQL server from your website.
22.	Write a program to read customer information like cust_no, cust_name, Item_purchase, and mob_no, from customer table and display all these information in table format on output screen.
23.	Write a program to edit name of customer to “Bob” with cust_no =1, and to delete record with cust_no=3.
24.	Write a program to read employee information like emp_no, emp_name, designation and salary from EMP table and display all this information using table format.
25.	Create a dynamic web site using PHP and MySQL.
	<b>56</b>

## DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

<b>II Year – II Semester</b>		<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>0</b>	<b>0</b>	<b>3</b>	<b>1.5</b>
<b>ADVANCED JAVA PROGRAMMING LAB</b>					

### Course Objectives:

The aim of this lab is to

- Practice programming in the Java
- Gain knowledge of object-oriented paradigm in the Java programming language
- Learn use of Java in a variety of technologies and on different platforms

### Course Outcomes:

By the end of the course student will be able to write java program for

- Evaluate default value of all primitive data type, Operations, Expressions, Control-flow, Strings
- Determine Class, Objects, Methods, Inheritance, Exception, Runtime Polymorphism, User defined Exception handling mechanism
- Illustrating simple inheritance, multi-level inheritance, Exception handling mechanism
- Construct Threads, Event Handling, implement packages, developing applets

## BEANS PROGRAMMING

1. Write a quiz applet and use gauge bean to update the score
2. Create a time zone list and retrieve any time which is given with zone using java beans
3. Develop a bean program that display a sequece of images in the form of slide show
4. Create a bean that displays a 3D plot of the following function  $Z = f(x,y) = 0.01 *(x^2 - y^2)$
5. Create a frame that instantiates the beans registers paints to receive color event notifications from selectors adds the beans to the frame and makes the frame visible
6. Create a bean that displays a pie chart and use pie customizer to update the pie chart
7. Develop a bean that takes date and year and represent it in the local language in the form of a calender For (Eg.) French , Italian etc

## SERVLETS PROGRAMMING

1. Write a servlet to display
  - a. IP address and Port no. of server
  - b. The host name and address of the computer on which your browser visits
2. Use a servlet as RMI client to enable a method given
3. Using servlet create a form which contain a text area, checkbox, radio button, label and text field with buttons
4. Create a chat program that uses servlets to communicate with 2 machines.
5. Create a servlet that gets the date and time of the system

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

<b>II Year – II Semester</b>		<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>0</b>	<b>0</b>	<b>2</b>	<b>1</b>
<b>R PROGRAMMING LAB</b>					

**OBJECTIVE:**

After taking the course, students will be able to Use R for statistical programming, computation, graphics, and modeling, Write functions and use R in an efficient way, Fit some basic types of statistical models Use R in their own research, Be able to expand their knowledge of R on their own.

**Prerequisites: R Programming, Statistics. List of Programs:**

1. Simple programs on R statements (if,if-else,switch,R Next, R Break, R For Loop, R Repeat)
2. Write a function to print squares of numbers in sequence.
3. Write a program to create a function with Arguments.
4. Write a Program to create a list with same data type and different data type.
5. Write a program to create two vectors of different lengths.
6. Write a program a) create a .csv file b) reading from .cse file c) analyzing the .cse file d) Writing in to the .csv file.
7. Write a program to create Pie charts, Bar Charts, Box Plot, Histogram, line Graphs using sample data.
8. Write a program to create input vector for lm () function using Linear Regression.
9. Write a program to create a sequence numbers between -1 and 20 incrementing by 0.2 Using Normal Distribution.
10. Write a program to create a sample of 100 numbers which are incremented by 1.5 using Binomial Distribution.
11. Write a program to create R time series object for a period of 12 month and plot it. ( consider the annual snowfall details at a place starting from Jan 2019 )

**REFERENCE BOOKS:**

- 1) R Cookbook, Paul Teetor, Oreilly.
- 2) R in Action, Rob Kabacoff, Manning

## DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

<b>II Year – I Semester</b>		<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>0</b>
<b>ESSENCE OF INDIAN TRADITIONAL KNOWLEDGE</b>					

### Course Objectives:

To facilitate the students with the concepts of Indian traditional knowledge and to make them understand the Importance of roots of knowledge system

- The course aim of the importing basic principle of third process reasoning and inference sustainability is at the course of Indian traditional knowledge system
- To understand the legal framework and traditional knowledge and biological diversity act 2002 and geographical indication act 2003
- The courses focus on traditional knowledge and intellectual property mechanism of traditional knowledge and protection
- To know the student traditional knowledge in different sector

### Course Outcomes:

After completion of the course, students will be able to:

- Understand the concept of Traditional knowledge and its importance
- Know the need and importance of protecting traditional knowledge
- Know the various enactments related to the protection of traditional knowledge
- Understand the concepts of Intellectual property to protect the traditional knowledge

### UNIT I

Introduction to traditional knowledge: Define traditional knowledge, nature and characteristics, scope and importance, kinds of traditional knowledge, the physical and social contexts in which traditional knowledge develop, the historical impact of social change on traditional knowledge systems. Indigenous Knowledge (IK), characteristics, traditional knowledge vis-à-vis indigenous knowledge, traditional knowledge Vs western knowledge traditional knowledge vis-à-vis formal knowledge

Learning Outcomes:

At the end of the unit, the student will able to:

- Understand the traditional knowledge.
- Contrast and compare characteristics importance kinds of traditional knowledge.
- Analyze physical and social contexts of traditional knowledge.
- Evaluate social change on traditional knowledge.

### UNIT II

Protection of traditional knowledge: the need for protecting traditional knowledge Significance of TK Protection, value of TK in global economy, Role of Government to harness TK.

Learning Outcomes:

At the end of the unit, the student will able to:

- Know the need of protecting traditional knowledge.
- Apply significance of tk protection.
- Analyze the value of tk in global economy.
- Evaluate role of government

### UNIT III

Legal framework and TK: A: The Scheduled Tribes and Other Traditional Forest Dwellers (Recognition of Forest Rights) Act, 2006, Plant Varieties Protection and Farmers Rights Act,



2001 (PPVFR Act);B:The Biological Diversity Act 2002 and Rules 2004, the protection of traditional knowledge bill, 2016. Geographical indications act 2003.

Learning Outcomes:

At the end of the unit the student will able to:

- Understand legal framework of TK.
- Contrast and compare the ST and other traditional forest dwellers
- Analyze plant variant protections
- Evaluate farmers right act

#### **UNIT IV**

Traditional knowledge and intellectual property: Systems of traditional knowledge protection, Legal concepts for the protection of traditional knowledge, Certain non IPR mechanisms of traditional knowledge protection, Patents and traditional knowledge, Strategies to increase protection of traditional knowledge, global legal FORA for increasing protection of Indian Traditional Knowledge.

Learning Outcomes:

At the end of the unit, the student will able to:

- Understand TK and IPR
- Apply systems of TK protection.
- Analyze legal concepts for the protection of TK.
- Evaluate strategies to increase the protection of TK.

#### **UNIT V**

Traditional knowledge in different sectors: Traditional knowledge and engineering, Traditional medicine system, TK and biotechnology, TK in agriculture, Traditional societies depend on it for their food and healthcare needs, Importance of conservation and sustainable development of environment, Management of biodiversity, Food security of the country and protection of TK.

Learning Outcomes:

At the end of the unit, the student will able to:

- Know TK in different sectors.
- Apply TK in engineering.
- Analyze TK in various sectors.
- Evaluate food security and protection of TK in the country.

#### **Reference Books:**

- 1) Traditional Knowledge System in India, by Amit Jha, 2009.
- 2) Traditional Knowledge System and Technology in India by Basanta Kumar Mohanta and Vipin Kumar Singh, PratibhaPrakashan 2012.
- 3) Traditional Knowledge System in India by Amit Jha Atlantic publishers, 2002
- 4) "Knowledge Traditions and Practices of India" Kapil Kapoor, Michel Danino

#### **esources:**

- 1) <https://www.youtube.com/watch?v=LZP1StpYEPM>
- 2) <http://nptel.ac.in/courses/121106003/>



**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY: KAKINADA**  
**KAKINADA – 533 003, Andhra Pradesh, India**