



**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA**  
**KAKINADA – 533 003, Andhra Pradesh, India**  
**DEPARTMENT OF CSE - COMPUTER SCIENCE & TECHNOLOGY**

## **COURSE STRUCTURE AND SYLLABUS**

**For UG – R20**

**B. Tech - COMPUTER SCIENCE & ENGINEERING with Specialization**

### **COMPUTER SCIENCE & TECHNOLOGY**

*(Applicable for batches admitted from 2020-2021)*



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



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

**DEPARTMENT OF CSE - COMPUTER SCIENCE & TECHNOLOGY**

**II Year – I SEMESTER**

S. No	Course Code	Course Title	L	T	P	C
1.	BSC2101	Mathematics – III	3	0	0	3
2.	PCC2101	Mathematical Foundations of Computer Science	3	0	0	3
3.	PCC2102	Data Structures	3	0	0	3
4.	PCC2103	Operating Systems	3	0	0	3
5.	PCC2104	Java Programming	3	0	0	3
6.	PCC2105	Data Structures Lab	0	0	3	1.5
7.	PCC2106	OS&UNIX Programming Lab	0	0	3	1.5
8.	PCC2107	Java Programming Lab	0	0	3	1.5
9.	SC2101	Free and Open Source Software	0	0	4	2
10.	MC2101	Essence of Indian Traditional Knowledge	2	0	0	0
<b>TOTAL</b>						<b>21.5</b>

**II Year – II SEMESTER**

S. No.	Course Code	Course Title	L	T	P	C
1.	ESC2201	Computer Organization & Architecture	3	0	0	3
2.	BSC2201	Probability and Statistics	3	0	0	3
3.	PCC2201	Formal Languages & Automata Theory	3	0	0	3
4.	PCC2202	Database Management System	3	0	0	3
5.	HSMC2201	Managerial Economics and Financial Accountancy	3	0	0	3
6.	ESC2202	Computer Organization & Architecture Lab	0	0	3	1.5
7.	PCC2203	R Programming Lab	0	0	3	1.5
8.	PCC2204	Database Management System Lab	0	0	3	1.5
9.	SC2201	Android Application Development	0	0	4	2
<b>TOTAL</b>						<b>21.5</b>
<b>Minor courses</b> (The hours distribution can be 3-0-2 or 3-1-0 also)			4	0	0	4
Internship 2 Months (Mandatory) during summer vacation						



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

**DEPARTMENT OF CSE - COMPUTER SCIENCE & TECHNOLOGY**

<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>MATHEMATICS - III (BSC2101)</b>					

**Course Objectives:**

- To familiarize the techniques in partial differential equations
- To furnish the learners with basic concepts and techniques at plus two level to lead them into advanced level by handling various real world applications.

**Course Outcomes:**

At the end of the course, the student will be able to

- Interpret the physical meaning of different operators such as gradient, curl and divergence (L5)
- Estimate the work done against a field, circulation and flux using vector calculus (L5)
- Apply the Laplace transform for solving differential equations (L3)
- Find or compute the Fourier series of periodic signals (L3)
- Know and be able to apply integral expressions for the forwards and inverse Fourier transform to a range of non-periodic waveforms (L3)
- Identify solution methods for partial differential equations that model physical processes (L3)

**UNIT I: Vector calculus:**

Vector Differentiation: Gradient – Directional derivative – Divergence – Curl – Scalar Potential.  
 Vector Integration: Line integral – Work done – Area – Surface and volume integrals – Vector integral theorems: Greens, Stokes and Gauss Divergence theorems (without proof).

**UNIT II: Laplace Transforms:**

Laplace transforms of standard functions – Shifting theorems – Transforms of derivatives and integrals – Unit step function – Dirac's delta function – Inverse Laplace transforms – Convolution theorem (without proof).

Applications: Solving ordinary differential equations (initial value problems) using Laplace transforms.

**UNIT III: Fourier series and Fourier Transforms:**

Fourier Series: Introduction – Periodic functions – Fourier series of periodic function-Dirichlet's conditions – Even and odd functions – Change of interval – Half-range sine and cosine series. Fourier Transforms: Fourier integral theorem (without proof) – Fourier sine and cosine integrals – Sine and cosine transforms – Properties – inverse transforms – Finite Fourier transforms.

**UNIT IV: PDE of first order:**

Formation of partial differential equations by elimination of arbitrary constants and arbitrary functions – Solutions of first order linear (Lagrange) equation and nonlinear (standard types) equations.



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

**DEPARTMENT OF CSE - COMPUTER SCIENCE & TECHNOLOGY**

**UNIT V: Second order PDE and Applications:**

Second order PDE: Solutions of linear partial differential equations with constant coefficients – RHS term of the type  $e^{ax+by}$ ,  $\sin(ax+by)$ ,  $\cos(ax+by)$ ,  $x^m y^n$ .

Applications of PDE: Method of separation of Variables – Solution of One dimensional Wave, Heat and two-dimensional Laplace equation.

**Text Books:**

- 1) B. S. Grewal, Higher Engineering Mathematics, 43<sup>rd</sup> Edition, Khanna Publishers.
- 2) B. V. Ramana, Higher Engineering Mathematics, 2007 Edition, Tata Mc. GrawHill Education.

**Reference Books:**

- 1) Erwin Kreyszig, Advanced Engineering Mathematics, 10<sup>th</sup> Edition, Wiley-India.
  - 2) Dean. G. Duffy, Advanced Engineering Mathematics with MATLAB, 3<sup>rd</sup> Edition, CRC Press.
  - 3) Peter O' Neil, Advanced Engineering Mathematics, Cengage.
- Srimantha Pal, S C Bhunia, Engineering Mathematics, Oxford University Press.



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

**DEPARTMENT OF CSE - COMPUTER SCIENCE & TECHNOLOGY**

<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>MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE(PCC2101)</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:**

At the end of the course student will be able to

- Demonstrate skills in solving mathematical problems
- Comprehend mathematical principles and logic
- Demonstrate knowledge of mathematical modeling and proficiency in using mathematical software
- Manipulate and analyze data numerically and/or graphically using appropriate software

Communicate effectively mathematical ideas/results verbally or in writing

**UNIT I**

Mathematical Logic: Propositional Calculus: Statements and Notations, Connectives, Well Formed Formulas, Truth Tables, Tautologies, Equivalence of Formulas, Duality Law, Tautological Implications, Normal Forms, Theory of Inference for Statement Calculus, Consistency of Premises, Indirect Method of Proof, Predicate Calculus: Predicates, Predicative Logic, Statement Functions, Variables and Quantifiers, Free and Bound Variables, Inference Theory for Predicate Calculus.

**UNIT II**

Set Theory: Sets: Operations on Sets, Principle of Inclusion-Exclusion, Relations: Properties, Operations, Partition and Covering, Transitive Closure, Equivalence, Compatibility and Partial Ordering, Hasse Diagrams, Functions: Bijective, Composition, Inverse, Permutation, and Recursive Functions, Lattice and its Properties, Algebraic Structures: Algebraic Systems, Properties, Semi Groups and Monoids, Group, Subgroup and Abelian Group, Homomorphism, Isomorphism.

**UNIT III**

Combinatorics: Basis of Counting, Permutations, Permutations with Repetitions, Circular and Restricted Permutations, Combinations, Restricted Combinations, Binomial and Multinomial Coefficients and Theorems, Number Theory: Properties of Integers, Division Theorem, Greatest Common Divisor, Euclidean Algorithm, Least Common Multiple, Testing for Prime Numbers, The Fundamental Theorem of Arithmetic, Modular Arithmetic, Fermat's and Euler's Theorems



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

**DEPARTMENT OF CSE - COMPUTER SCIENCE & TECHNOLOGY**

**UNIT IV**

Recurrence Relations: Generating Functions, Function of Sequences, Partial Fractions, Calculating Coefficient of Generating Functions, Recurrence Relations, Formulation as Recurrence Relations, Solving Recurrence Relations by Substitution and Generating Functions, Method of Characteristic Roots, Solving Inhomogeneous Recurrence Relations.

**UNIT V**

Graph Theory: Basic Concepts, Graph Theory and its Applications, Sub graphs, Graph Representations: Adjacency and Incidence Matrices, Isomorphic Graphs, Paths and Circuits, Eulerian and Hamiltonian Graphs, Multigraphs, Bipartite and Planar Graphs, Euler's Theorem, Graph Colouring and Covering, Chromatic Number, Spanning Trees, Prim's and Kruskal's Algorithms, BFS and DFS Spanning Trees.

**Text Books:**

- 1) Discrete Mathematical Structures with Applications to Computer Science, J. P. Tremblay and P. Manohar, Tata McGrawHill.
- 2) Elements of Discrete Mathematics-A Computer Oriented Approach, C. L. Liu and D. P. Mohapatra, 3<sup>rd</sup> Edition, Tata McGrawHill.

**Reference Books:**

- 1) Discrete Mathematics for Computer Scientists and Mathematicians, J. L. Mott, A. Kandel and T. P. Baker, 2<sup>nd</sup> Edition, Prentice Hall of India.
- 2) Discrete Mathematical Structures, Bernard Kolman, Robert C. Busby and Sharon Cutler Ross, PHI.
- 3) Discrete Mathematics and its Applications with Combinatorics and Graph Theory, K. H. Rosen, 7<sup>th</sup> Edition, Tata McGrawHill.

**e-Resources:**

- 1) <https://nptel.ac.in/courses/106/106/106106094/>



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

**DEPARTMENT OF CSE - COMPUTER SCIENCE & TECHNOLOGY**

<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>DATA STRUCTURES (PCC2102)</b>					

**Course Objectives:**

The objective of the course is to

- Introduce the fundamental concept of data structures and abstract datatypes
- 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 datatypes
- 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.



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

**DEPARTMENT OF CSE - COMPUTER SCIENCE & TECHNOLOGY**

**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 Prim's & Kruskal's 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. Forouzan, Cengage.
- 3) Data Structures with C, Seymour Lipschutz TMH

**e-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)





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

**DEPARTMENT OF CSE - COMPUTER SCIENCE & TECHNOLOGY**

<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>OPERATING SYSTEMS (PCC2103)</b>					

**Course Objectives:**

The objectives of this course is to

- Introduce to the internal operation of modern operating systems
- Define, explain, processes and threads, mutual exclusion, CPU scheduling, deadlock, memory management, and filesystems
- Understand File Systems in Operating System like UNIX/Linux and Windows
- Understand Input Output Management and use of Device Driver and Secondary Storage (Disk) Mechanism
- Analyze Security and Protection Mechanism in Operating System

**Course Outcomes:**

After learning, the course the students should be able to:

- Describe various generations of Operating System and functions of Operating System
- Describe the concept of program, process and thread and analyze various CPU Scheduling Algorithms and compare their performance
- Solve Inter Process Communication problems using Mathematical Equations by various methods
- Compare various Memory Management Schemes especially paging and Segmentation in Operating System and apply various Page Replacement Techniques
- Outline File Systems in Operating System like UNIX/Linux and Windows

**UNIT I**

Operating Systems Overview: Operating system functions, Operating system structure, Operating systems operations, Computing environments, Open-Source Operating Systems.

System Structures: Operating System Services, User and Operating-System Interface, systems calls, Types of System Calls, system programs, operating system structure, operating system debugging, System Boot.

**UNIT II**

Process Concept: Process scheduling, Operations on processes, Inter-process communication, Communication in client server systems.

Multithreaded Programming: Multithreading models, Thread libraries, Threading issues.

Process Scheduling: Basic concepts, Scheduling criteria, Scheduling algorithms, Multiple processor scheduling, Thread scheduling.

Inter-process Communication: Race conditions, Critical Regions, Mutual exclusion with busy waiting, Sleep and wakeup, Semaphores, Mutexes, Monitors, Message passing, Barriers, Classical IPC Problems - Dining philosophers problem, Readers and writers problem.



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

**DEPARTMENT OF CSE - COMPUTER SCIENCE & TECHNOLOGY**

**UNIT III**

Memory-Management Strategies: Introduction, Swapping, Contiguous memory allocation, Paging, Segmentation.

Virtual Memory Management: Introduction, Demand paging, Copy on-write, Page replacement, Frame allocation, Thrashing, Memory-mapped files, Kernel memory allocation.

**UNIT IV**

Deadlocks: Resources, Conditions for resource deadlocks, Ostrich algorithm, Deadlock detection and recovery, Deadlock avoidance, Deadlock prevention.

File Systems: Files, Directories, File system implementation, management and optimization.

Secondary-Storage Structure: Overview of disk structure, and attachment, Disk scheduling, RAID structure, Stable storage implementation.

**UNIT V**

System Protection: Goals of protection, Principles and domain of protection, Access matrix, Access control, Revocation of access rights.

System Security: Introduction, Program threats, System and network threats, Cryptography for security, User authentication, implementing security defenses, Firewalling to protect systems and networks, Computer security classification.

Case Studies: Linux, Microsoft Windows.

**Text Books:**

- 1) Silberschatz A, Galvin P B, and Gagne G, Operating System Concepts, 9th edition, Wiley, 2013.
- 2) Tanenbaum A S, Modern Operating Systems, 3rd edition, Pearson Education, 2008. (for Interprocess Communication and Filesystems.)

**Reference Books:**

- 1) Dhamdhare D M, Operating Systems A Concept Based Approach, 3rd edition, Tata McGraw-Hill, 2012.
- 2) Stallings W, Operating Systems - Internals and Design Principles, 6th edition, Pearson Education, 2009
- 3) Nutt G, Operating Systems, 3rd edition, Pearson Education, 2004.

**e-Resources:**

- 1) <https://nptel.ac.in/courses/106/105/106105214/>



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

**DEPARTMENT OF CSE - COMPUTER SCIENCE & TECHNOLOGY**

<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 (PCC2104)</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,



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

**DEPARTMENT OF CSE - COMPUTER SCIENCE & TECHNOLOGY**

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 AdjustersClass.

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 StringBuilder.

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.
3. JAVA Programming for Core and Advanced Learners, Sagayaraj, Denis, Karthik, Gajalakshmi, Universities Press.



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

**DEPARTMENT OF CSE - COMPUTER SCIENCE & TECHNOLOGY**

**References Books:**

1. Introduction to java programming, 7th edition by Y Daniel Liang, Pearson
2. Constructive JAVA Programming, Sagayaraj, Denis, Karthik, Gajalakshmi, Universities Press.

**e-Resources:**

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)



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

**DEPARTMENT OF CSE - COMPUTER SCIENCE & TECHNOLOGY**

<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>DATA STRUCTURES LAB(PCC2105)</b>					

**Course Objectives:**

The objective of this lab is to

- Demonstrate procedural and object oriented paradigm with concepts of streams, classes, functions, data and objects.
- Understand dynamic memory management techniques using pointers, constructors, destructors, etc
- Demonstrate the concept of function overloading, operator overloading, virtual functions and polymorphism, inheritance.
- Demonstrate the different data structures implementation.

**Course Outcomes:**

By the end of this lab the student is able to

- Use basic data structures such as arrays and linked list.
- Programs to demonstrate fundamental algorithmic problems including Tree Traversals, Graph traversals, and shortest paths.
- Use various searching and sorting algorithms.

**Exercise 1:**

Write recursive program which computes the  $n^{\text{th}}$  Fibonacci number, for appropriate values of  $n$ . Analyze behavior of the program Obtain the frequency count of the statement for various values of  $n$ .

**Exercise 2:**

Write recursive program for the following

- a) Write recursive and non recursive C program for calculation of Factorial of an integer
- b) Write recursive and non recursive C program for calculation of GCD ( $n, m$ )
- c) Write recursive and non recursive C program for Towers of Hanoi :  $N$  disks are to be transferred from peg  $S$  to peg  $D$  with Peg  $I$  as the intermediate peg.

**Exercise 3:**

- a) Write C program that use both recursive and non recursive functions to perform Linear search for a Key value in a given list.
- b) Write C program that use both recursive and non recursive functions to perform Binary search for a Key value in a given list.
- c) Write C program that use both recursive and non recursive functions to perform Fibonacci search for a Key value in a given list.

**Exercise 4:**

- a) Write C program that implement Bubble sort, to sort a given list of integers in ascending order
- b) Write C program that implement Quick sort, to sort a given list of integers in ascending order
- c) Write C program that implement Insertion sort, to sort a given list of integers in ascending order

**Exercise 5:**

- a) Write C program that implement heap sort, to sort a given list of integers in ascending order
- b) Write C program that implement radix sort, to sort a given list of integers in ascending order
- c) Write C program that implement merge sort, to sort a given list of integers in ascending order

**Exercise 6:**

- a) Write C program that implement stack (its operations) using arrays
- b) Write C program that implement stack (its operations) using Linked list



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

**DEPARTMENT OF CSE - COMPUTER SCIENCE & TECHNOLOGY**

**Exercise 7:**

- a) Write a C program that uses Stack operations to Convert infix expression into postfix expression
- a) Write C program that implement Queue (its operations) using arrays.
- b) Write C program that implement Queue (its operations) using linked lists

**Exercise 8:**

- a) Write a C program that uses functions to create a singly linked list
- b) Write a C program that uses functions to perform insertion operation on a singly linked list
- c) Write a C program that uses functions to perform deletion operation on a singly linked list

**Exercise 9:**

- a) Adding two large integers which are represented in linked list fashion.
- b) Write a C program to reverse elements of a single linked list.
- c) Write a C program to store a polynomial expression in memory using linked list
- d) Write a C program to represent the given Sparse matrix using arrays.
- e) Write a C program to represent the given Sparse matrix using linked list

**Exercise 10:**

- a) Write a C program to Create a Binary Tree of integers
- b) Write a recursive C program for Traversing a binary tree in preorder, inorder and postorder.
- c) Write a non recursive C program for Traversing a binary tree in preorder, inorder and postorder.
- d) Program to check balance property of a tree.

**Exercise 11:**

- a) Write a C program to Create a BST
- b) Write a C program to insert a node into a BST.
- c) Write a C program to delete a node from a BST.



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

**DEPARTMENT OF CSE - COMPUTER SCIENCE & TECHNOLOGY**

<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>OS &amp; UNIX PROGRAMMING LAB (PCC2106)</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 Banker's Algorithm for Dead Lock Avoidance
- 9) Simulate Banker's Algorithm for Dead Lock Prevention.
- 10) Simulate the following page replacement algorithms:  
 a) FIFO b) LRU c) LFU





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

**DEPARTMENT OF CSE - COMPUTER SCIENCE & TECHNOLOGY**

- 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.



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

**DEPARTMENT OF CSE - COMPUTER SCIENCE & TECHNOLOGY**

<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 (PCC2107)</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)**

- a) Write a JAVA program to display default value of all primitive data type of JAVA
- b) 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.
- c) 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)**

- a) Write a JAVA program to search for an element in a given list of elements using binary search mechanism.
- b) Write a JAVA program to sort for an element in a given list of elements using bubblesort
- c) Write a JAVA program to sort for an element in a given list of elements using mergesort.
- d) Write a JAVA program using StringBuffer to delete, remove character.

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

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

**Exercise - 4 (Methods)**

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

**Exercise - 5 (Inheritance)**

- a) Write a JAVA program to implement Single Inheritance
- b) 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?



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

**DEPARTMENT OF CSE - COMPUTER SCIENCE & TECHNOLOGY**

**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 classpath
- 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.



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

**DEPARTMENT OF CSE - COMPUTER SCIENCE & TECHNOLOGY**

II Year - I Semester		L	T	P	C
		0	0	4	2

**FREE AND OPEN SOURCE SOFTWARE (SC2101)**

**Course Objectives:**

The student should be made to:

- Be exposed to the context and operation of free and open source software (FOSS) communities and associated software projects.
- Be familiar with participating in a FOSS project
- Learn scripting language like Python or Perl
- Learn programming language like Ruby
- Learn some important FOSS tools and techniques

1. Getting started with Linux basic commands and directory structure, execute file, directory operations.

2. Linux commands for redirection, pipes, filters, job control, file ownership, filepermissions, links and file system hierarchy.

3. Shell Programming : Write shell script to show various system configuration like

- 1 Currently logged user and his logname
- 2 Your current shell
- 3 Your home directory
- 4 Your operating system type
- 5 Your current path setting
- 6 Your current working directory
- 7 Show Currently logged number of users

4. Write shell script to show various system configuration like

- 1 About your OS and version, release number, kernel version
- 2 Show all available shells
- 3 Show mouse settings
- 4 Show computer CPU information like processor type, speed etc
- 5 Show memory information
- 6 Show hard disk information like size of hard-disk, cache memory, mode etc
- 7 File system (Mounted)

5. Shell script program for scientific calculator.

6. Version Control System setup and usage using GIT.

- 1 Creating a repository
- 2 Checking out a repository
- 3 Adding content to the repository
- 4 Committing the data to a repository



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

**DEPARTMENT OF CSE - COMPUTER SCIENCE & TECHNOLOGY**

7. Shell script to implement a script which kills every process which uses more than aspecified value of memory or CPU and is run upon system start.
8. Running PHP : simple applications like login forms after
9. Advanced linux commands curl, wget, ftp, ssh and grep
10. Application deployment on a cloud-based LAMP stack/server with PHP eg: Openshift, Linode etc.
11. Virtualization environment (e.g., xen, kqemu, virtualbox or lguest) to test anapplications, new kernels and isolate applications. It could also be used to exposestudents to other alternate OSs like \*BSD
12. Introduction to packet management system : Given a set of RPM or DEB, how to buildand maintain, serve packages over http or ftp. and also how do you configure clientsystems to access the package repository.
13. Installing various software packages. Either the package is yet to be installed or anolder version is existing. The student can practice installing the latest version. Of course,this might need Internet access.
  - 1 Install samba and share files to windows
  - 2 Install Common Unix Printing System(CUPS)

**TEXT BOOK:**

Ellen Siever, Stephen Figgins, Robert Love, Arnold Robbins, “Linux in a Nutshell”, Sixth Edition, OReilly Media, 2009.

**REFERENCES:**

Philosophy of GNU URL: <http://www.gnu.org/philosophy/>.

Linux Administration URL: <http://www.tldp.org/LDP/lame/LAME/linux-admin-made-easy/>.

The Python Tutorial available at <http://docs.python.org/2/tutorial/>.



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

**DEPARTMENT OF CSE - COMPUTER SCIENCE & TECHNOLOGY**

<b>II Year - I Semester</b>		<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>2</b>	<b>0</b>	<b>0</b>	<b>0</b>
<b>ESSENCE OF INDIAN TRADITIONAL KNOWLEDGE (MC2101)</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 knowledgesystem.
- To understand the legal framework and traditional knowledge and biological diversity act 2002 and geographical indication act2003.
- The courses focus on traditional knowledge and intellectual property mechanism of traditional knowledge andprotection.
- To know the student traditional knowledge in differentsector.

**Course Outcomes:**

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

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

**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

**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.

**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.

**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.



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

**DEPARTMENT OF CSE - COMPUTER SCIENCE & TECHNOLOGY**

**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.

**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

**e-Resources:**

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