

**DATA COMMUNICATION AND NETWORKING**

**LAB MANUAL**

**SCHOOL OF COMPUTING SCIENCE & ENGINEERING**

**COMPUTER SCIENCE & ENGINEERING 2022-2023**



|  |  |  |  |
| --- | --- | --- | --- |
| **SUBJECT** | **Java Programming Lab** | **PROGRAMME** | **B. Tech.** |
| **SUBJECT CODE** | **BCSE2333** | **SEMESTER** | **3** |
| **CREDITS** | **2** | **DURATION OF**  **SEMESTER** | **13 Weeks** |
| **PREREQUISITE**  **SUBJECTS** | **Java** | **SESSION**  **DURATION** | **2 Hrs per Week** |

# Vision

"To be recognized globally as a premier School of Computing Science and Engineering for imparting quality and value based education within a multi-disciplinary and collaborative research based environment."

# Mission

## The mission of the school is to:

**M1:** Develop a strong foundation in fundamentals of computing science and engineering with responsiveness towards emerging technologies.

**M2:** Establish state-of-the-art facilities and adopt education 4.0 practices to analyze, develop, test and deploy sustainable ethical IT solutions by involving multiple stakeholders.

**M3:** Foster multidisciplinary collaborative research in association with academia and industry through focused research groups, Centre of Excellence, and Industry Oriented R&D Labs.

# PROGRAM EDUCATIONAL OBJECTIVES

## The Graduates of Computer Science and Engineering shall:

**PEO1:** be engaged with leading Global Software Services and Product development companies handling projects in cutting edge technologies.

**PEO2:** serve in technical or managerial roles at Government firms, Corporates and contributing to thesociety as successful entrepreneurs through startup.

**PEO3:** undertake higher education, research or academia at institutions of transnational reputation.

# PROGRAMME SPECIFIC OUTCOME (PSO):

## The students of Computer Science and Engineering shall:

**PSO1:** Have the ability to work with emerging technologies in computing requisite to Industry 4.0.

**PSO2:** Demonstrate Engineering Practice learned through industry internship and research project tosolve live problems in various domains.

# PROGRAMME OUTCOME (PO):

**PO1 Computing Science knowledge:** Apply the knowledge of mathematics, statistics, computing science and information science fundamentals to the solution of complex computer application problems.

**PO2 Problem analysis:** Identify, formulate, review research literature, and analyze complex computing science problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and computer sciences.

**PO3 Design/development of solutions:** Design solutions for complex computing problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

**PO4 Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

**PO5 Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern computing science and IT tools including prediction and modeling to complex computing activities with an understanding of the limitations.

**PO6 IT specialist and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional computing science and information science practice.

**PO7 Environment and sustainability**: Understand the impact of the professional computing science solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

**PO8 Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the computing science practice.

**PO9 Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

**PO10 Communication:** Communicate effectively on complex engineering activities with the IT analyst community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

**PO11 Project management and finance:** Demonstrate knowledge and understanding of the computing science and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

**PO12 Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

# List of Programs



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr. No. | | Title of Lab Experiments | | |
| 1 | | a) Write a JAVA program to print “Hello World.”  b)WAP in java to print the values of various primitive data types. | | |
| 2 | | WAP in java to demonstrate operator precedence. | | |
| 3 | | WAP in java to create a class Addition and a method add() to add two numbers. | | |
| 4 | | Write a program that uses length property for displaying any number of command line arguments. | | |
| 5 | | WAP in java to find the average of N numbers. | | |
| **6** | | WAP in java to find out factorial of a number | | |
| 7 | | WAP in java to find out the Fibonacci series | | |
| 8 | | WAP in java to sort elements in 1D array in ascending order. | | |
| 9 | | WAP in java to perform matrix addition using two 2D arrays. | | |
| 10 | | WAP in java to find out the number of characters in a string. | | |
| 11 | | WAP in java that implements method overloading. | | |
| 12 | | Create a class Shape and override area() method to calculate area of rectangle, square and circle | | |
| 13 | | WAP in java to demonstrate the properties of public private and protected variables and methods (with package). | | |
| 14 | | WAP that illustrates method overriding | | |
| 15 | | a) WAP in java demonstrating arithmetic exception and ArrayOutOf Bounds Exception using try catch block  b) WAP in java to demonstrate the use of nested try block and nested catch block. | | |
| 16 | | Write a program to count the number of times a character appears in a File.  [Note : The character check is case insensitive... i.e, ‘a’ and ‘A’ are considered to be the same] | | |
| 17 | | WAP in java in two different ways that creates a thread by (i) extending thread class (ii) implementing runnable interface which displays 1st 10 natural numbers. | | |
| 18 | | a) WAP in java that connects to a database using JDBC & insert values into table.  b) WAP to connect to a database using JDBC and delete values from table. | | |
| **Value Added List of Experiments**   1. Create TreeSet, HashSet, LinkedHashSet. Add same integer values in all three. Display all three set and note down the difference. 2. WAP in java WAP in java and run applications that use "List" Collection objects 3. Create a HashMap<Rollno, Name>. Display Map, take rollno from user and display name. 4. WAP to do CRUD operation in java   **Name & Signature of Course Coordinator:** | |  |



**Course Outcomes**



Upon successful completion of this course, students will be able to

|  |  |
| --- | --- |
| **CO1** | Understand the java platform, structure of java class and java packages. |
| **CO2** | Understand object oriented concepts and implement the same using java operators, control statements, arrays. |
| **CO3** | Apply inheritance and exception handling concepts in solving problems. |
| **CO4** | Apply java IO stream concepts to solve problems efficiently. |
| **CO5** | Implement multithreading, collections and Java database connectivity concepts to solve problems in advance level. |
| **CO6** | Apply the concept of JDBC for connecting the application to the database. |

**CO-PO-PSO MAPPING:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CO/PO**  **Mapping**  (S/M/W or 3/2/1 indicates strength of correlation) S/3-Strong, M/2-Medium, L/1-Low | | | | | | | | | | | | | | |
| CO’s | **Programme Outcomes (POs) / Programme Specific Outcome (PSO)** | | | | | | | | | | | | | |
| PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 |
| CO1 | 2 |  | 1 |  | 2 |  |  |  |  |  |  | 2 | 2 | 2 |
| CO2 | 2 |  |  |  | 2 |  |  |  |  |  |  | 1 | 1 |  |
| CO3 | 2 | 2 |  | 2 | 2 |  |  |  |  |  |  | 1 | 2 | 3 |
| CO4 | 2 |  |  | 1 | 2 |  |  |  |  |  |  |  | 2 |  |
| CO5 | 2 |  |  | 1 |  |  |  |  |  |  |  | 1 | 1 |  |
| CO6 | 2 | 2 |  |  |  |  |  |  |  |  |  | 1 | 2 |  |

**Continuous Assessment Pattern**

|  |  |  |
| --- | --- | --- |
| **Practical IA** | **ETE** | **Total** |
| 100 | 0 | 100 |

**Rubrics for Practical IA**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Rubrics - Parts** | **Marks** |
| 1 | Performance | **2** |
| 2 | Result | **3** |
| 3 | File | **2** |
| 4 | Viva | **3** |
| **Total** | | **10** |

**Name of the Course Coordinator Signature**