PARVATHANENI BRAHMAYYA
SIDDHARTHA COLLEGE OF ARTS & SCIENCE
Autonomous
Siddhartha Nagar, Vijayawada–520010
Re-accredited at 'A+' by the NAAC

<table>
<tr><td colspan="4"><b>Course Code</b></td><td colspan="4"><b>23AIMAP233</b></td></tr>
<tr><td colspan="4"><b>Title of the Course</b></td><td colspan="4"><b>Object Oriented Programming using Java Lab</b></td></tr>
<tr><td colspan="4"><b>Offered to: (Programme/s)</b></td><td colspan="4">B.Sc. Honours (Artificial Intelligence)</td></tr>
<tr><td><b>L</b></td><td><b>0</b></td><td><b>T</b></td><td><b>0</b></td><td><b>P</b></td><td><b>2</b></td><td><b>C</b></td><td><b>1</b></td></tr>
<tr><td colspan="2"><b>Year of Introduction:</b></td><td colspan="2"><b>2024-25</b></td><td colspan="3"><b>Semester:</b></td><td><b>3</b></td></tr>
<tr><td colspan="2"><b>Course Category:</b></td><td colspan="2"><b>Major Lab</b></td><td colspan="2"><b>Course Relates to:</b></td><td colspan="2"><b>GLOBAL</b></td></tr>
<tr><td colspan="2"><b>Year of Revision:</b></td><td colspan="2"><b>--</b></td><td colspan="2"><b>Percentage:</b></td><td colspan="2"><b>--</b></td></tr>
<tr><td colspan="4"><b>Type of the Course:</b></td><td colspan="4"><b>Employability</b></td></tr>
<tr><td colspan="4"><b>Crosscutting Issues of the Course :</b></td><td colspan="4"><b>-</b></td></tr>
<tr><td colspan="4"><b>Pre-requisites, if any</b></td><td colspan="4"><b>Programming Knowledge</b></td></tr>
</table>

Course Description:
This lab course focuses on practical Java programming skills. Students will explore type casting, sorting, and object-oriented concepts such as class creation, method overloading, and constructors. They will work on matrix multiplication, inheritance types, and runtime polymorphism. The course includes multi-threading, exception handling, and package management, providing hands-on experience with error handling and concurrent programming. Additionally, students will learn to create applets and manage file operations, including reading and writing files. This course offers a thorough understanding of Java programming through diverse, real-world applications and challenges.

Course Aims and Objectives:

S.NO	COURSE OBJECTIVES
1	**Learn** to use basic Java programming to handle type casting, sorting, and data operations.
2	**Create** Java classes and methods, including constructors, to perform calculations and manage data.
3	**Develop** programs for matrix operations, different types of inheritance, and interfaces.
4	**Understand** and use multi-threading and exception handling in Java, and work with packages.
5	**Practice** creating applets and managing file operations, including reading and writing files.

Course Outcomes

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

CO NO	COURSE OUTCOME	BTL	PO	PSO
CO1	**Apply** type casting and sorting techniques in Java to process and organize data effectively.	**K3**	**PO2, PO5, PO6**	**PSO1,PSO2**
CO2	**Design** and **implement** classes, methods, and constructors to perform calculations and manage object attributes in Java.	**K6**	**PO2, PO3, PO5, PO6**	**PSO1,PSO2**
CO3	**Develop** Java programs to demonstrate various types of inheritance, matrix operations, and interface implementations.	**K6**	**PO2, PO3, PO5, PO6**	**PSO1,PSO2**
CO4	**Analyze** and **handle** multi-threading and exception scenarios, including creating threads, managing exceptions, and demonstrating package usage.	**K4**	**PO2, PO3, PO5, PO6**	**PSO2**
CO5	**Construct** and **manage** applets and file operations, including writing and reading files, to create interactive applications and handle file data.	**K6**	**PO2, PO3, PO5, PO6**	**PSO1,PSO2**

For BTL: K1: Remember; K2: Understand; K3: Apply; K4: Analyze; K5: Evaluate; K6: Create

CO-PO MATRIX									
CO NO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2
CO1	1	-	-	-	2	2	-	2	1
CO2	2	-	2	-	2	3	-	2	1
CO3	-	-	2	-	3	3	2	2	3
CO4	-	-	2	-	3	3	3	-	1
CO5	-	-	2	-	2	3	-	2	1

Use the codes 3,2,1 for High, Moderate and Low correlation Between CO-PO-PSO respectively

Course Structure

This lab list covers the key areas of Object Oriented Programming in Java Lab course, providing hands-on practice

Unit-1: Introduction to Java Programming (6Hrs)

Lab 1

1. Design Java program to perform Type Casting in java.
   - **Dataset (web link) / Experiment:** Java Type Casting Example and
   - **Tasks:** Demonstrate type casting with different data types, including integer to double and string to integer conversions.
2. Develop a Java program for sorting a given list of names in ascending order.
   - **Dataset (web link) / Experiment:** Java Sorting Example
   - **Tasks:** Sort a list of names in ascending order using Java's built-in sorting methods.

Unit-2: Control statements, Classes, Objects and Methods (6Hrs)

Lab 2

3. Create a class Rectangle. The class has attributes length and width. It should have methods that calculate the perimeter and area of the rectangle. It should have read Attributes method to read length and width from user.
    - **Dataset (web link) / Experiment:** Java Rectangle Class Example
    - **Tasks:** Define Rectangle class with methods to calculate perimeter, area, and read attributes (length and width) from user.
4. Construct a Java program that implements method overloading.
    - **Dataset (web link) / Experiment:** Java Method Overloading Example
    - **Tasks:** Implement method overloading by defining multiple methods with the same name but different parameters for various operations.

Lab 3

5. Write a program to demonstrate Parameterized Constructors.
    - **Dataset (web link) / Experiment:** Explore constructor usage in Java.
    - **Tasks:** Create a class with a parameterized constructor to initialize attributes with given values and demonstrate its application.
6. Write a Program to demonstrate Constructor Overloading.
    - **Dataset (web link) / Experiment:** Understand constructor overloading in Java.
    - **Tasks:** Implement multiple constructors within a class, each with different parameter lists to illustrate how constructor overloading works.

Unit-3: Inheritance, Arrays, Strings and Interfaces (6Hrs)

Lab 4

7. Design a Java program to calculate multiplication of 2 matrices.
    - **Dataset (web link) / Experiment:** Explore matrix operations in Java.
    - **Tasks:** Implement a program to perform matrix multiplication using nested loops and display the resultant matrix.
8. Develop a Java program for extending and implementing interfaces.
    - **Dataset (web link) / Experiment:** Understand interfaces in Java.
    - **Tasks:** Create an interface and a class that implements this interface. Extend the class to demonstrate interface inheritance and implementation.

Lab 5

9. Construct Java programs to implement various types of inheritance
   i. Single ii. Multi-Level iii. Hierarchical iv. Hybrid
    - **Dataset (web link) / Experiment:** Explore different inheritance types in Java.
    - **Tasks:** Implement Java programs to demonstrate single, multi-level, hierarchical, and hybrid inheritance using appropriate class structures.
10. Develop a java program to implement runtime polymorphism.
    - **Dataset (web link) / Experiment:** Learn about runtime polymorphism in Java.
    - **Tasks:** Create a program demonstrating runtime polymorphism by using method overriding and dynamic method dispatch with base and derived classes.

Lab 6

11. Write a program to demonstrate Abstract Classes and Final Keyword
    - **Dataset (web link) / Experiment:** Understand abstract classes and final keyword in Java.
    - **Tasks:** Create a Java program using abstract classes and methods, and apply the final keyword to classes and methods to prevent inheritance and modification.
12. Design a program for extending and implementing interfaces.
    - **Dataset (web link) / Experiment:** Explore interfaces in Java.

- **Tasks:** Develop a Java program that defines and implements multiple interfaces, demonstrating how to extend and use interface methods in a class.

Unit-4: Multi-Threading, Exception Handling and Packages (6Hrs)

Lab 7

13. Write a Java program which accepts withdraw amount from the user and throws an exception "In Sufficient Funds" when withdraw amount more than available amount.
    - **Dataset (web link) / Experiment:** Explore exception handling in Java.
    - **Tasks:** Write a Java program to accept a withdrawal amount from the user and throw an "Insufficient Funds" exception if the withdrawal amount exceeds the available balance.
14. Construct a Java program to create three threads and that displays "good morning", for every one second, "hello" for every 2 seconds and "welcome" for every 3 seconds by using extending Thread class.
    - **Dataset (web link) / Experiment:** Explore multi-threading in Java.
    - **Tasks:** Create a Java program with three threads that display "good morning" every 1 second, "hello" every 2 seconds, and "welcome" every 3 seconds by extending the Thread class.

Lab 8

15. Construct a Java program that creates three threads. First thread displays "OOPS", the second thread displays "Through" and the third thread Displays "JAVA" by using Runnable interface.
    - **Dataset (web link) / Experiment:** Explore thread creation using the Runnable interface.
    - **Tasks:** Write a Java program to create three threads where the first thread displays "OOPS", the second thread displays "Through", and the third thread displays "JAVA" using the Runnable interface.
16. Design a program to create and Import Packages
    - **Dataset (web link) / Experiment:** Explore Java package creation and import.
    - **Tasks:** Create a Java program to define and import custom packages.

Lab 9

17. Construct Java program to implement various types of Exception Handling Mechanisms
    i. Arithmetic Exception
    ii. Number Format Exception
    iii. ArrayIndexOutofBounds Exception
    - **Dataset (web link) / Experiment:** Explore different exception handling mechanisms.
    - **Tasks:** Construct a Java program to demonstrate handling:
        - Arithmetic Exception
        - Number Format Exception
        - ArrayIndexOutOfBounds Exception
18. Design a program to demonstrate Catch Blocks
    - **Dataset (web link) / Experiment:** Explore the usage of catch blocks in exception handling.
    - **Tasks:** Design a Java program to demonstrate the use of catch blocks for handling exceptions.

Unit-5: Streams, Files, Graphic Programming (6Hrs)

Lab -10

19. Develop a program for writing and reading Files

- **Dataset (web link) / Experiment:** Explore file I/O operations.
- **Tasks:** Develop a Java program to write data to a file and read it back.

20. Write a java program to Draw a circle with a radius of 50 pixels.
    - **Dataset (web link) / Experiment:** Explore graphics programming with Java.
    - **Tasks:** Write a Java program to draw a circle with a radius of 50 pixels using the Graphics class.

References:

1. E.Balaguruswamy, Programming with JAVA, A primer, 3e, TATA McGraw-Hill Company.

Web Resources:

- Prof.DebasisSamanta, Dept of Computer science, IIT Kharagpur.“Basic Concepts of Java Programming”, 2018.

https://www.youtube.com/watch?v=OjdT2l-EZJA&list=PLfn3cNtmZdPOe3R_wO_h540QNfMkCQ0ho&index=1

***

Lab Model Question Paper Pattern

23AIMAP233:Object Oriented Programming using Java Lab

Offered to: B.Sc. Honours (Artificial Intelligence)

Max. Marks: 50 **Max. Time: 3Hrs**

Pass. Min: 20

(A)	**Evaluation Procedure**	**35 Marks**
I	Experiments (Exam & Execution)	**30 Marks**
II	Viva	**3 Marks**
III	Record	**2 Marks**
(B)	**CONTINUOUS ASSESMENT(Internal)**	**15 MARKS**

15 marks for the continuous assessment (Day to day work in the laboratory shall be evaluated for 15 marks by the concerned laboratory teacher based on the regularity/ record/viva). Laboratory teachers are mandated to ensure that every student completes 80%-90% of the lab assessments.

TOTAL: (A)+(B) = **50 MARKS**