RAJAGIRI SCHOOL OF ENGINEERING AND TECHNOLOGY (AUTONOMOUS)
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

S3 CSE

102802/CO300A OBJECT ORIENTED PROGRAMMING CONCEPTS USING JAVA

LAB CYCLE 2025

(A) Basic programs using data types, operators, control statements , array in Java , searching and sorting

Day 1

1. Develop a Java program that takes the student's score as input and prints their corresponding grade. Use an else-if ladder to define the grading criteria (e.g., A for scores 90 and above, B for scores between 80 and 89, etc.).
2. Write a Java program that takes an integer as input and reverses its digits using a for loop.
3. Develop a menu-driven calculator program in Java using a do-while loop. The program should display a menu with options for addition, subtraction, multiplication, and division. It should keep executing until the user chooses to exit.

Day 2

4. Write a Java program to manage class attendance and perform search and sort operations.
    - Read the roll numbers of students who attended the class. Using linear search, check if a student attended the class(using roll no) and display appropriate messages.
    - Sort the roll numbers of students present in ascending order and display the sorted list. (use bubble sort)
    - Display absentees roll numbers
5. Develop a Java program to read two matrices, display their sum and also find the transpose of the resultant matrix.

(B) Programs using object-oriented programming concepts to understand the use of constructors, inheritance, method overloading, method overriding and polymorphism.

Day 3

6. Create a Java class for a bank account. Include attributes like the account number, account holder's name, and balance. Implement methods to deposit money, withdraw money, check balance and display the account details. Create 2 objects and initialize their parameters using different types of constructors.
7. Create a class hierarchy for animals, including a base class "Animal," a subclass "Herbivore," and another subclass "Carnivore." Animal class should have attributes like name, age, and subclass should have the attribute type of animal. Use the super keyword to call the

constructor of the superclass when creating objects of subclasses. Demonstrate hierarchical inheritance and attribute initialization with the super keyword. Include a method in subclasses to display the details of the animals.

Day 4, Day 5

8. Design a class hierarchy to represent different roles within a company using object-oriented programming principles.

   i. Define an abstract base class "Employee" that includes common attributes such as name, employee_id, and salary, along with abstract methods like calculate_bonus() and display_info().

   ii. Create sub classes such as "Manager", "Developer", and "Intern" that inherit from the Employee. Each sub class should:

   a. Implement the abstract methods from the Employee.
   b. Add role-specific attributes and methods (e.g., team_size for Manager, programming_languages for Developer, duration for Intern).

   iii. Extend the hierarchy by introducing multi-level inheritance as follows:

   a. Create a Senior Manager class that inherits from Manager, and a Junior Developer class that inherits from Developer.
   b. These sub classes should override methods and include additional attributes (e.g., stock_options for SeniorManager, mentor for JuniorDeveloper).

   iv. Demonstrate method overloading by creating overloaded versions of a method such as assign_task() in the Developer class to handle different types or numbers of parameters.

   v. Create instances of each class and call their methods to show polymorphism and inheritance in action.

(C) Programs handling different types of files as well as input and output management methods.

Day 6

9. Write a program to read numbers in a file and copy the prime numbers among them into another file.
10. Implement a simple file encryption program that reads a text file, encrypts its contents, and then saves the encrypted data to a new file. You can use a simple substitution cipher, such as shifting characters by a fixed number.

(D) Exception handling and multi-threading applications:

Day 7

11. Write a Java program to implement an ATM for State Bank of India. The customers of the bank can withdraw money, deposit money, and transfer money through the ATM. Customers can also view the available balance. The account balance of all customers is stored in a file having a file name same as their account number. Throw an exception during a withdrawal operation if sufficient balance is not available and use a catch statement to inform the customer about the same. Use the finally block to ensure that the available balance is displayed and that the file that stores the account balance of the customer is closed after every transaction even if an exception occurs or not during the transaction. Handle all the possible exceptions related to file.

Day 8

12. Write a Java program that shows thread synchronization and multithreading. Create a deposit thread and a withdraw thread to simulate deposit and withdrawal operations on a shared bank account. Both threads repeatedly call the same function updateBalance() that is defined in a class BankAccount. The updateBalance() function accepts an integer as its argument: a positive value indicates a deposit and a negative value indicates a withdrawal. The function prints the current balance after the update operation twice, once immediately after the update and once after the thread sleeps for 1500ms. The function should be synchronized to ensure that only one thread accesses the shared resource (balance) at a time. The deposit thread calls the updateBalance() method with positive values, and the withdraw thread calls it with negative values.

E. Programs using Graphical user interface, event handling and database support

Day 9

13. Write a Java program that works as a simple calculator to calculate the area of a circle, area of rectangle, and area of triangle. Add a text field to display the result. Handle any possible exceptions like divide by zero.
14. Write a Java program to create a GUI based application that contains two text fields and one button. The application receives an integer through the first text field, checks if that integer is a perfect number or not and displays the result in the second text field when the button "Check" is clicked. If the number is not a perfect number, then display the sum of the digits of that number.

Day 10

15. The empid, name, designation, nationality, year of joining and salary of all the employees of a Multinational Company are stored in a table. Write a Java program using JDBC (i) to display the details of all the employees in the descending order of their salary (ii) to display the details of a given employee using empid **(use GUI)**, (iii) to display the details of the employees of a given nationality and a given designation (iv) to delete the details of the employees who joined before a given year.

Lab in Charges:

Mr. Biju Abraham N. (S3 CS Alpha)

Ms. Amitha Mathew (S3 CS Beta)

Mr. Sebin Jose (S3 CS Gamma)

Mr. Sandy Joseph (S3 CS Delta)

HoD:

Dr. Preetha K. G.