RAMAIAH
Institute of Technology

Ramaiah Institute of Technology
(Autonomous Institute, Affiliated to VTU)
Department of Information Science and Engineering

Question for the Laboratory: Semester End Examination (SEE)

Course Name: Advanced Java Lab	Term: March 2025 to June 2025
Course Code: ISL48	IA Marks: 50
Credits: 0:0:1	Exam Hours: 03, Max. Marks: 50

Question No.	Program Description
1.	Write a Java Program for the following Payroll System. Create a package called PayrollManagement. Define an interface Payable with a method calculate(). Create a super class called Employee with the following data members name, age, salary and designation. Use a parameterised constructor to initialize all the data members. The Employee class is inherited in three sub-classes namely FullTime, PartTime and Intern and these subclasses implement the Payable interface. Full-Time Employee receives monthly salary and Part-Time Employee receives hourly wages. Intern takes a fixed stipend. Create another package "Company" and import these classes to complete the payroll system. Create an **ArrayList** of Employee objects in the **Company package.**
2.	Create a Queue class **to include strings(Name)** and implement enqueue, dequeue and display operations. Create Custom Exceptions to handle Queue Overflow and Queue Underflow.
3.	Define a package named stringoperations to encapsulate the string processing functionality. Create an interface named StringManipulator with the following methods for string operations: • String reverse(String input): Reverses the given string. • String toUpperCase(String input): Converts the string to uppercase. • String concatenate(String str1, String str2): Concatenates two strings. • int countVowels(String input): Counts the number of vowels in the string. • int wordCount(String input): Counts the number of words in the string. Implement the interface in a class named StringProcessor within the stringoperations package, providing concrete implementations for all methods. Create a main class in a separate package **Application**, where: • An object of StringProcessor is created. • The user is prompted to enter a string. • Each method of the interface is tested and its output is displayed.
4.	Create a Palindrome Checker program using StringBuffer that verifies if a given string is a palindrome. The program should implement user-defined exception handling for the following: 1. If the input string contains **special** characters, throw a custom exception InvalidInputException. 2. If the string length is **greater than 5** characters, throw a custom exception LongStringException. 3. If the string is a palindrome (irrespective of the case), print a message indicating that it is a palindrome.

	4. If the string is not a palindrome, print a message indicating that it is not a palindrome.
5.	Create a Password Security Application in Java that takes a user's password as input and performs the following operations . o Check if the password contains at least one uppercase letter, one lowercase letter, and one digit. ● **Count the number of characters (other than the special characters)** o Count the number of special characters in the password. o Mask the password by replacing all characters with * (except the first and last character). o Reverse the password and display it (for security encryption demonstration). o Append a random security token (e.g., "@123!") to the password. o Replace all vowels (a, e, i, o, u) with # to make it difficult to read.
6.	Write a Telephone Program to do the following. **Create a LinkedList which is a collection of already available contacts (Name, contact number)** . ● Accept missed calls from the user. For each missed call, store the time of call, telephone number of origin, and name of the caller if the name is available. For unlisted numbers, set the name to "private caller". Store this into an **ArrayList**. The ArrayList can hold at most 5 caller details. On receiving the 6th call, replace with the oldest call in the arraylist. ● Iterate through the missed call list. Prompt the user to indicate whether to display the details of the call or delete the call from the ArrayList. ● Include an option to display the contact list and the missed call list.
7.	Write a Java program using user-defined storage classes to create a book database and store it in a Collection (**Hash Map**) List. a. Books collection should include title, author, publisher and price. b. Write a method to sort the books in ascending order of price and store it in another List. Maintain the book details with respect to a unique book id. c. Prompt for an author name and list all the books with the same author name. Create a new list holding all the book details with price greater than a user specified price.
8.	Write a program to create generic Stack class with push(), pop(), clear(), isEmpty() and display() methods. **Use an ArrayList to save the contents of Stack.** Demonstrate creating Stack of String and **Float** objects.
9.	Create a desktop java application using swings to enable a user to enter student information such as name, usn, age, address, sgpa of 4 semesters, category(**as combobox)**. a. Perform validations on all the fields. Display appropriate messages in pop up boxes to indicate wrong entries. b. On clicking of the "compute" button, find the cgpa (Average of sgpas) . On clicking of the "done" button, place the student details (name, usn, age, address,cgpa, category) in a collection. c. Display the collection in a textarea on the click of a "complete" button. Enable / Disable the visibility of "compute" and "done" buttons.
10.	Write a java program using Swing to validate user login information using dialog boxes. a. Once validated, allow the user to enter the customer id, if the person is a new customer, else check whether the customer exists in a collection and obtain the customer id. b. The customer id can be obtained given a mobile number. Allow the user to enter the item purchased by giving the item id and quantity purchased.

	c. On clicking of a button, the item name and the total cost should appear in the corresponding GUI components. d. Using option dialog box, indicate the types of discount available for the customer. On clicking on the print button, print the details in information dialog box.
11.	Write a program that uses Java Swing and JDBC to create a stand-alone application: a. Create two tables namely, Representative (RepNo, RepName, State, Comission, Rate) and Customer (CustNo, CustName, State, Credit_Limit, RepNo) in MySQL database. Use appropriate Swing components to insert values in a form. b. Use another form to display Representative's information when Credit_Limit is above 15,000.
12.	Create a JSP based web application for purchasing a type of shirt from a list of shirts. Display a table giving the type of shirts and cost. With a drop down menu, select the type of shirt. Use radio-buttons to select the neck type of shirt (Round neck, V neck and collar). In a textbox, enter the quantity. Include "compute" button that calculates the total cost of purchase done and displays the purchase details. Include appropriate labels.

Evaluation Criteria (Max. Marks:50)	
Write Up	08 Marks
Conduction and Execution	35 Marks
Viva	07 Marks
Change of Program	Write up is evaluated for only 3 marks

Course Co-ordinator **HoD**