CHRIST (Deemed to be University), Bangalore – 560 029

Department of Computer Science

CIA - Component -II Repeat Practical Test – November 2024

PG 2 Trimester

Programme Name: MCA **Max. Marks: 30**

Course Name: Java Programming **Time: 2 Hrs**

Course Code: MCA272

General Instructions

- All rough work should be done in the answer script. Do not write or scribble in the question paper except your register number.
- Verify the Course code / Course title & number of pages of questions in the question paper.
- Make sure your mobile phone is switched off and placed at the designated place in the hall.
- Malpractices will be viewed very seriously.
- Answers should be written on both sides of the paper in the answer booklet. No sheets should be detached from the answer booklet.
- Answers without the question numbers clearly indicated will not be valued. No page should be left blank in the middle of the answer booklet.
-

Course Outcomes (COs): The students will able to

CO1: Understanding and applying the principles of object-oriented programming in the construction of robust, maintainable programs.

CO2: Analyse the various societal and environmental problems critically to develop solutions using the features of programming language.

CO3: Develop sustainable and innovative solutions for real-time problems.

Answer all the questions: **Total Marks: 30**

Q. No	Questions	CO	RBT
1	**Marks 15** The financial agency aims to simulate an application that tracks and manages banking tools and their interactions with various financial products. Implement the following classes and interfaces: Classes and Interfaces **Abstract Class: BankingTool** Private attribute: toolName (String) Protected method: setToolName(String name) Public method: getToolName() **Interface: Audit** Method: void performAudit()	CO2 & CO3	L3, L4 & L5

	Interface: ProductManagement Method: void manageProduct(String productType) **Class: SoftwareTool (extends BankingTool and implements Audit)** Attribute: softwareType (String) Constructor: Initializes toolName and softwareType Override: performAudit() method to print a message about auditing the software tool **Class: LoanProcessingTool (extends SoftwareTool and implements ProductManagement)** Attribute: securityProtocol (String) Constructor: Initializes toolName, softwareType, and securityProtocol Override: manageProduct(String productType) method to print a message about managing the given product type using the loan processing tool **Task** Write the complete implementation of the above classes and interfaces in Java, ensuring the proper usage of access modifiers (private, protected, and public). **Test the Program with the following scenario:** Create an instance of the LoanProcessingTool class. Set the tool name using the setToolName method. Print the tool name and manage a financial product (e.g., "Home Loan") using the loan processing tool. Call the performAudit method for the tool.		
2	**Barking Dog (7.5 Marks)** We have a dog that likes to bark. We need to wake up if the dog is barking at night! Write a method shouldWakeUp that has 2 parameters. 1st parameter should be of type boolean and be named barking it represents if our dog is currently barking. 2nd parameter represents the hour of the day and is of type int with the name hourOfDay and has a valid range of 0-23. We have to wake up if the dog is barking before 8 or after 22 hours so in that case return true. In all other cases return false. If the hourOfDay parameter is less than 0 or greater than 23 return false.	CO1, CO2 & CO3	L3, L4 L5, L6

	Examples of input/output: shouldWakeUp (true, 1); → should return true shouldWakeUp (false, 2); → should return false since the dog is not barking. shouldWakeUp (true, 8); → should return false, since it's not before 8. shouldWakeUp (true, -1); → should return false since the hourOfDay parameter needs to be in a range 0-23. TIP: Use the if else statement with multiple conditions. NOTE: The shouldWakeUp method needs to be defined as public static like we have been doing so far in the course. NOTE: Do not add a main method to solution code.		
3	**Minimum Element (7.5 Marks)** **Write a method called readInteger() that has no parameters and returns an int.** **It needs to read in an integer from the user - this represents how many elements the user needs to enter.** **Write another method called readElements() that has one parameter of type int** **The method needs to read from the console until all the elements are entered, and then return an array containing the elements entered.** **And finally, write a method called findMin() with one parameter of type int[]. The method needs to return the minimum value in the array.** **The scenario is:** **1. readInteger() is called.** **2. The number returned by readInteger() is then used to call readElements().** **3. The array returned from readElements() is used to call findMin().** **4. findMin() returns the minimum number.**		

	[Do not try and implement this. It is to give you an idea of how the methods will be used] **TIP: Assume that the user will only enter numbers, never letters.** **TIP: Instantiate (create) the Scanner object inside the method. There are two scanner objects, one for each of the two methods that are reading in input from the user.** **TIP: Be extremely careful about spaces in the printed message.** **NOTE: All methods should be defined as private static.** **NOTE: Do not add a main method to the solution code.** **NOTE: Classes that are not in the java.lang package should be manually imported.**		

Revised Bloom's Taxonomy (RBT) Levels :		
L1 – Remembering	L2 – Understanding	L3 – Applying
L4 – Analyzing	L5 – Evaluating	L6 - Creating