Examination	: B.Tech Semester IV	Seat No.	: 82
Date	: 04/01/2024	Day	: THURSDAY
Time	: 1:00 PM TO 2:15 PM	Max. Marks	: 36

INSTRUCTIONS:

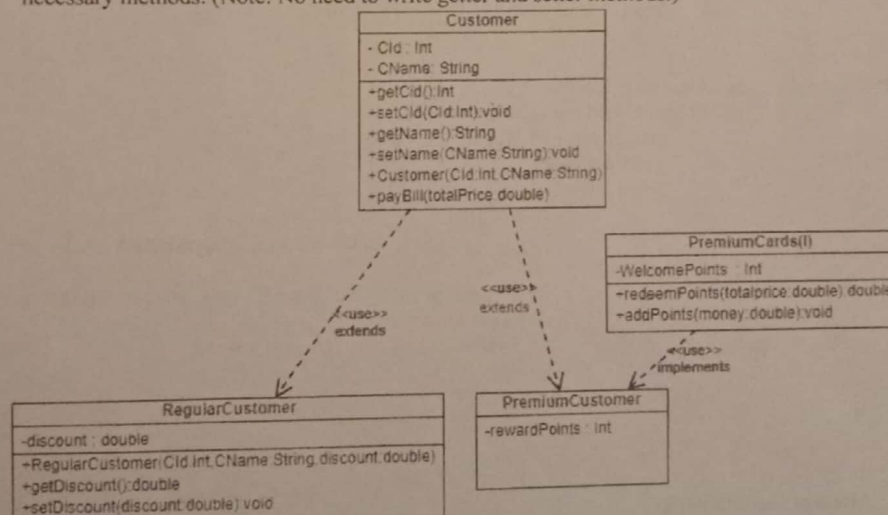| 1. | Figures to the right indicate maximum marks for that question. |
2.	The symbols used carry their usual meanings.
3.	Assume suitable data, if required & mention them clearly.
4.	Draw neat sketches wherever necessary.

Q.1 Do as directed. [12]

CO1-R (a) What is a checked exception? Demonstrate a checked exception with an appropriate code. [2]

CO1-U (b) Match the following [2]

Method	State Transition
sleep()	Runnable/Running to Blocked
wait()	Runnable/Running to waiting
join()	New to Runnable
start()	Blocked to Running

CO1-A (c) Demonstrate how an object reference passed as an argument to a method. Is it pass-by-value or pass-by-reference? [2]

CO1-R (d) State true or false and justify your answer. [2]
   i. JVM is platform independent.
   ii. Java is a Robust language.

CO1-U (e) Differentiate static block vs. instance initializer block. [2]

CO1-N (f) There is a package p1 which contains class Demo, package p2 which contains class Demo. Is it possible to access the Demo class of package p1 and p2 in class A of package p3? If yes, explain the syntax to access the Demo class of package p1 and package p2, if No then justify your answer. [2]

Q.2 Attempt Any TWO from the following: [12]

CO1-C (a) The TechFoods company uses java in order to keep track of their customer details and payment details. It has a **Customer** class which is used to store the details of the customers. The customers are of two types **RegularCustomer** and **Premium Customer**. Only the Regular customers are provided with a discount of 5% on the total cost. Premium customers are special types of regular customers they will get an extra 5% discount. Premium customers also have membership cards and hence must also implement the interface **PremiumCards**. All the Premium customers will get 100 welcome points (100 points = 100 rs.). Implement necessary methods. (Note: No need to write getter and setter methods.) [6]

CO1-C (b) Analyze the following code fragment and answer the questions. [6]

```java
public class Container<T>{
    private T value;
    public Container(T value) {
        this.value = value;
    }
}
```

```java
public class Container2<T> extends Container<T>{
    private T value2;
    public Container2(){}
    public Container2(T value1,T value2) {
        super(value1);
        this.value2 = value2;
    }
}
```

How will Java react to attempts to compile and run these lines? Justify your answer.

   i)   Container<String> myContainer1 = new Container<>();
       What is T and String in the context of Container class?

   ii)   Container myContainer2 = new Container("Good morning!");
       What is myContainer2 in this context? Explain.

   iii) Container<String> myContainer1=new Container2<>("Java","Technologies");
       Container2<String> myContainer2=myContainer1;

CO1-C   **(c)**   **(I)** What will be the output of the following code snippet?   **[4]**

class DemoStrings{
 public static void main(String []args){
 String s1 = "JT"
 s1 = "Java Technologies";
 System.out.println("The value of s1=" + s1);
 String s2 = "JT";
 System.out.println("The value of s2=" + s2);
 System.out.println(s1.equals(s2));
 System.out.println(s1 == s2);
 String s3 = s2;
 System.out.println("The value of s3=" + s3);
	StringBuffer sb1 = new StringBuffer(s3);
	 sb1.append(",");
	 sb1.append("new");
	 System.out.println("The value of sb1=" +
	 sb1.toString());
	 System.out.println(s3.equals(sb1.toString()));
	 String s4 = sb1.toString();
	 String []a = s4.split(",")
	 for(String x : a){
	 System.out.println("value="+x);
	 }}}

CO1-C   **(II)** What will be the output of the following code snippet?   **[2]**

class A{
   public static void main(String []args){
     int a[] = {1,2,3,4,5};
     for(int x : a)
       x+=10;
	for(int i=0; i<a.length; i++)
	       System.out.println(a[i]);
	    }
	 }

Q.3   **Attempt the following:**   **[12]**

CO1-N   **(a)**   State if the following statements are true or false.   **[6]**

   i.   A nested class lowers the level of encapsulation.

   ii. Anonymous inner class is declared as a subclass of some nameless superclass.

   iii. A local inner class can be private.

   iv. The static nested class cannot access non-static data members.

   v. Member Inner class can access any private instance variable of the outer class.

   vi. Outer.Inner obj = new Outer.Inner(); Here, obj can be an object of static nested class.

CO1-E   **(b)**   Fill in the blanks and rewrite the entire program. Provide the output generated by this code.   **[6]**

```
class AgeException_____Throwable {
        public AgeException(String message) {
                _____(message);
        }
        _____ String toString() {
                return "AgeException [" + _____.getMessage() + "]";
        }
}
public class DemoAgeException {
        public static void main(String args[]) {
                int age[] = { 15, 45, 10, 33, 18 };
                for (int a : age) {
                        try {
                                if (a < 18){
                                        _____ new AgeException("inappropriate age to vote");
                                }
                                System.out.println("At " + a + " years, you are eligible to vote");
                        } catch (AgeException ae) {
                                System.out.println("Caught " + ae);
                        }
                }
        }
}
```

OR

Q.3   **Attempt the following:**   **[12]**

CO1-A   **(a)**   Enlist all the uses of static keyword and demonstrate any two in a suitable Java program.   **[6]** Also, mention the output that your code is expected to generate.

CO1-C   **(b)**   Create two classes ThreadA and ThreadB using Runnable interface. ThreadA will be   **[6]** responsible for printing the square of the number and ThreadB will be responsible for printing the cube of the number. Print squares and cubes of numbers from 1 to 5 using these classes. Ensure that the main thread finishes last.

Bloom's Taxonomy levels: R-Remembering, U- Understanding, A-Applying, N-Analyzing, E- Evaluating, C-Creating