राष्ट्रीय प्रौद्योगिकी संस्थान जमशेदपुर
NATIONAL INSTITUTE OF TECHNOLOGY JAMSHEDPUR
(An Institution of National Importance under MoE, Government of India)

Department of Computer Science and Engineering

Semester: Autumn Semester 2024-25
Examination: Mid Semester
Course Title: Java Programming
Course Code: CS3305
Full Marks: 30
Duration: 2 Hrs.
Semester: MCA 3rd Semester
Faculty: Dr. Mayukh Sarkar

(Answer All Questions)

1. Consider the following java file.

```
package midsem.pkg1;

public class A {
   public class B {
      B() {}
   }
}
```

(a) If you compile this java file, mention the names of the class files generated within the project. **[1]**
(b) How will you create an object of class B in another class, say Main, outside of class A? **[1.5]**
(c) If the signature of class B is changed to *static,* how will you create the object of B in that case? **[1.5]**
(d) If the Main class resides in another package, you will receive an error message while creating the object. What will be the error? How will you resolve it? **[2]**

2. (a) How will you create a thread as anonymous inner class? Elaborate with an example of a program, where the child thread gets created as anonymous inner class, printing from 100 to 120 with 500 ms sleep in between each print, and at the same time the main thread will print from 1 to 10, then will wait for the child thread to exit, and then will print from 11 to 20 again. **[4]**
(b) Suppose you are creating an object, say *obj,* of some third-party library class, and you do not have access to the source code. The developer has not made the class thread-safe, but in a part of your program, you need only one thread to access *obj* at any time. How will you achieve that? **[2]**

3. (a) Let *Employee* be an interface containing some abstract method *assignProject()*. This interface has two concrete classes *Developer* and *Maintainer*. Write brief Java codes for *Employee, Developer* and *Maintainer.* Create a factory method in *Employee* to return one of the two subclass objects and elaborate Dynamic Method Dispatch via this program. **[4]**
(b) Now, if *Developer* class has some new method *getProgress()* which is not in *Employee*, will you be able to access the method from the variable corresponding to *Developer* object in part (a) Dynamic Method Dispatch code? If yes, elaborate with the invocation of the method via the variable, and if no, what error will you receive and how will you rectify the error? **[2]**

4. (a) An array of n integers needs to be populated with the cubes from 1 to n. Write a multithreaded program that will perform the job in multiple threads of 8 threads. Your program must consider the fact that n may not be multiple of 8. [3]
(b) Demonstrate the effect of *final* keyword at variable, method and class levels, with suitable examples. [3]

5. (a) Demonstrate the differences between checked and unchecked exceptions by creating manual exceptions of both types. Demonstrate the use of *finally* keyword with suitable example. [2+1]
(b) Let *MyException* be a user-created unchecked exception. Suppose this exception got generated during the execution of a program, but the original reason of this exception was some invalid argument to a method, and you want to assign *IllegalArgumentException* with the message "Wrong arguments" as main reason of the *MyException* thrown. How will you perform this assignment. Give a brief overview of how JVM will show the exception handling message if not caught in the program. [3]

राष्ट्रीय प्रौद्योगिकी संस्थान जमशेदपुर
NATIONAL INSTITUTE OF TECHNOLOGY JAMSHEDPUR
(An Institution of National Importance under MoE, Government of India)
Department of Computer Science and Engineering

Semester: Autumn Semester 2024-25
Course Title: Java Programming
Full Marks: 50
Semester: MCA 3rd Semester

Examination: End Semester
Course Code: CS3305
Duration: 3 Hrs.
Faculty: Dr. Mayukh Sarkar

(Answer All Questions)

1. **(a)** Demonstrate how Encapsulation, Polymorphism and Inheritance are implemented in Java, with suitable examples. **[3]**
(b) What do you mean by Method Overloading and Method Overriding? Demonstrate with suitable examples. **[2+2]**
(c) How will you pass an object as a parameter to a method in Java? Demonstrate with a suitable example with its behavior from the perspective of JVM Heap memory. **[3]**

2. **(a)** Let *Child* be some subclass of another class *Parent*. Parent class has a constructor with an integer parameter, and Child class does not have any constructor. "JVM will throw an exception while creating an object of Child" – is it a correct statement? If yes, demonstrate why? If not, make the statement correct and demonstrate the correct statement. **[2]**
(b) Now, add a protected member, say 'a' to Parent class, and place Child class to a different package. Now access the variable from Child class as follows.

```
Parent p = new Parent();
p.a = 10;
```

This code will throw an error. Why? In such case, how will you correctly access 'a' from Child? **[2]**
(c) Let *Inner* be a nested class inside *Outer* class. Within nested class, create four different variables of four different access specifiers, and do the same in the outer class. Now, try to access the inner variables from outer class, and outer variables from inner class. Demonstrate the accessibility of the variables in the stated scenario. **[3]**
(d) Let *A* be some subclass of class *B*. Both the classes have three initializers – constructor, static initializer and non-static initializer. Now, if you create an object of class *A*, what will be sequence of initializer invocations. Demonstrate with suitable example. **[3]**

3. **(a)** Let *V* be a class that contains a single variable *x*. Two classes, *P* and *Q* share a single object of class V, say *obj*. P updates the value of *obj.x* one by one from 1 to 10, whereas Q needs to consume the value of *obj.x* after each update by P. However, the objects of P and Q must perform this in parallel in two separate threads. "Write by P and consumption by Q will not be synchronized even after making the write and consume methods synchronous" – Demonstrate the statement with suitable implementations of the classes and show an example sequence of access after making the methods *synchronized*. How will you achieve the task while keeping the execution of P and Q in parallel? Elaborate on the solution. **[4+3]**

(b) Given a text file named 'test.txt' containing two integer numbers separated by a single space, on the same line. Write a Java program using try-with-resources that will read the line via *BufferedReader*, do the necessary processing and print its sum on the Console using *PrintWriter*. **[3]**

4. **(a)** What is a container in Java Swing? Name four top-level containers. **[2]**
(b) Write a Java program using swing that will create a window of width 300 and height 300. It will have a label, a button group of radio buttons with options *Windows* and *Linux*, and a clickable button with text *Submit*. On clicking *Submit*, the label will have the text "Submitted", and on selecting one of the radio buttons, it will show the corresponding text (Windows or Linux) in the label. At a time only one radio button can be selected. **[4]**
(c) Demonstrate the use and event handling of Toggle Button & Checkbox in Java Swing, via a suitable example of a frame containing both of these elements along with a label and handling suitable events on Toggle Button press and Checkbox select/unselect. **[4]**

5. **(a)** Consider a database with a table named student (id int primary key auto_increment, name varchar(50), age int, address varchar(50)). Consider a suitable Connection object named "conn" has been provided in the JDBC program. Perform the following tasks. **[2.5 x 4 = 10]**
   - **(i)** Insert a new row with three values provided by user for (name, age, address) in the order mentioned.
   - **(ii)** Modify the age of the student to 22, whose name value is provided by user.
   - **(iii)** Print the names of all students whose age is less that the value provided by user.
   - **(iv)** Delete the row from the table whose name and address are provided by user.