

Indian Institute of Technology Roorkee

End-term Examination (CLOSED BOOK)

Course: Fundamentals of Object Oriented Programming (CSN-103)

Date: Feb 02, 2023

Duration: 3 hr.

Max. Marks: 40

Q1:[3 Marks] What is the key difference between an abstract class and an interface having default methods? Consider two interfaces in the code below, each with a default method *display()*:

```
interface Employee {
    default void display() {
        System.out.println("Executing display in Employee");
    }
}

interface Customer {
    default void display() {
        System.out.println("Executing display in Customer");
    }
}
```

Write a Java program that defines a class **Bank** implementing both *Employee* and *Customer* interfaces. **Bank** class should contain the *main()* method. Create an object of **Bank** in the *main()* and call the *display()* method for the object such that the *display()* defined in *Customer* gets executed. Write additional statements and methods in the **Bank** class, if necessary.

Q2:[2 Marks] Give a detailed UML diagram for all the interfaces and classes used in Q1 along with their relationships. Mention all the methods used with their access specifiers.

Q3:[3 Marks] What restrictions does the **final** keyword impose when it is used with a) Variables b) Methods c) Classes?

Q4:[1 Mark] [MCQ] Which of the following lines of code will NOT compile if replaced with the commented line in the below code?

```
public interface PrintMe{
    // Add your code here
}
```

- | | | |
|-------------------------------------|---------------------------------|-------------------------------------|
| a) public static final int id = 10; | b) public static final id = 10; | c) public final static int id = 10; |
| d) public static final int = 10; | e) public static id = 10; | f) public static final id; |

Q5:[2.5 Marks] In what situations, Java display an error related to ambiguous call to an overloaded variable length argument (vararg)? Does the code below execute successfully? If not, correct **only** the statements causing the error(s).

```
class Ambiguity {
    static void Display(int ... a) {
        for(int i=0;i<a.length;i++)
            System.out.println("Element: "+ a[i]);
    }
    static void Display(boolean ... a) {
        for(int i=0;i<a.length;i++)
            System.out.println("Boolean Value: "+ a[i]);
    }
    public static void main(String args[]) {
        Display(2,3,4,5,6);
        Display(true,false,true);
    }
}
```

Q6:[2.5 Marks] Fill in the blanks (YES or NO) in the following table considering the class member access. For example: If a class member is **private**, it can be accessed by other code within the **same class** → YES

CLASS MEMBER ACCESS	No Modifier	Public	Private	Protected
Same Class			YES	
Same Package and Subclass				
Same Package and Non-subclass				
Different Package and Subclass				
Different Package and Non-subclass				

Q7:[2 Marks] Consider two packages b1 and b2. Both packages contain one class Box and nothing else. Does the code snippet given below execute without any error? If no, then point out the erroneous statement(s) and correct them. If yes, then printBox() of which Box class will be executed?

```
import b1.*;
import b2.Box.*;
class SampleProg {
    public static void main(String arg[]) {
        Box b = new Box();
        b.printBox();
    }
}
```

Q8:[1 Mark] [MCQ] Which of these are supporting by Java:

- a) Multiple Inheritance of classes
- b) Multi-level Inheritance of classes
- c) Multiple Inheritance of Interfaces
- d) Multi-level Inheritance of Interfaces

Q9: [1 Mark] [MCQ] What is polymorphism in Java?

- a) The ability of an object to take on many forms
- b) The ability of a class to inherit data and behavior from a superclass
- c) The ability of a subclass to override methods from a superclass
- d) None of the above

Q10: [1 Mark] [MCQ] What is the relationship between inheritance and polymorphism in Java?

- a) Polymorphism is a way to implement inheritance
- b) Inheritance provides the foundation for polymorphism
- c) They are independent concepts
- d) None of the above

Q11: [1 Mark] [MCQ] What does UML stand for?

- a) Unified Markup Language
- b) Unified Modeling Language
- c) Universal Modeling Language
- d) Universal Markup Language

Q12:[20 Marks] True or False (Don't write explanation)

- a) A user program can have either **catch** or **finally** block but not both.
- b) A subclass can call a protected member method of its superclass belonging to a different package.
- c) `import abc.xyz.*;` will import all the classes in the package **xyz** but not the sub-packages of **xyz**.
- d) A subclass can access both public and private members of the superclass.
- e) Overriding occurs only when the names and the type signatures of the two or more methods are identical.
- f) A reference of the superclass can be used to call a method of subclass only if the method is overridden.
- g) We cannot create an object of abstract class but can create an object reference.
- h) A reference variable of a user-defined class can refer to an instance of Object class.
- i) A protected method in the super-class can be private in the subclass.
- j) Final methods can be overridden in the derived class.
- k) Static methods and private methods cannot be overridden in the derived class.
- l) Overriding methods can have different return types.
- m) Static method does not have access to the instance variable directly (without passing an object).
- n) We cannot overload two methods if they differ only by static keyword.
- o) Data/elements within a final array can be changed/manipulated.
- p) We can pass an array as a variable length argument to a method.
- q) It is not possible to catch and handle multiple exceptions in a single catch block in Java.
- r) Java interface can have private methods.
- s) It is mandatory for a non-abstract class implementing an interface to provide an implementation for all its methods.
- t) Overloaded method can have different return types.

--END--