PROGRAMMING IN JAVA

Assignment 4

TYPE OF QUESTION: MCQ

Number of questions: 10 Total marks**: 10 × 1 = 10**

QUESTION 1:

Which of these access specifiers must be used for `main()` method?

a. **private**

b. **public**

c. **protected**

d. **default**

Correct Answer:

b. **public**

Detailed Solution:

`main()` method must be specified `public` as it called by Java run time system, outside of the program. If no access specifier is used then by default member is `public` within its own package & cannot be accessed by Java run time system.

QUESTION 2:

Consider the program given below.

```
public class Main{
    public static void main(String args[]){
        System.out.println(cos(2*PI));
    }
}
```

What will be the output if the program is executed?

a. **It will give compile-time error**

b. **It will give run-time error**

c. **1.0**

d. **3.14**

Correct Answer:

b. **It will give compile-time error**

Detailed Solution:

The program gives a compile time error as the Math class is missing.
The static import statement needs to be used to import the static members (e.g., PI) of java.lang.Math.

```
import static java.lang.Math.*;
```

This will allow the program to use PI directly.

QUESTION 3:

Consider the following code:

```
class Person {
    int a = 1;
    int b = 0;
    public Person() {
        System.out.println(a * b + " Java " );
    }
}

class Employee extends Person {
    int a = 0;
    int b = 1;
    public Employee() {
        System.out.println(a + b + " Java " );
    }
}

public class Question {
    public static void main(String args[]) {
        Person p = new Employee();
    }
}
```

What will be the output of the code given above?

a. **1 Java 10**
   0 Java 0

b. **1 Java**
   0 Java

c. **10 Java**
   0 Java

d. **0 Java**
   1 Java

Correct Answer:

d. **0 Java**
   1 Java

Detailed Solution:

If no super() or this() is included explicitly within the derived class constructor, the super() is implicitly invoked by the compiler. Therefore, in this case, the Person class constructor is called first and then the Employee class constructor is called. Up casting is allowed.

QUESTION 4:

In Java, can a subclass in a different package access a superclass's protected method?

a. **Yes, without any restrictions.**

b. **Yes, but only if they are in the same package.**

c. **No, protected methods are not accessible by subclasses.**

d. **Yes, but only through inheritance (i.e., using super or this, not via an object)**

Correct Answer:

d. **Yes, but only through inheritance (i.e., using super or this, not via an object)**

Detailed Solution:

- A subclass in a different package can access protected members only through inheritance (directly via super or this).

- It cannot access the superclass's protected method via an instance of the superclass.

- Protected also allows access to classes in the same package without subclassing, but that's not the case here.

QUESTION 5:

A package is a collection of:

a. **Only classes**

b. **Only interfaces**

c. **editing tools**

d. **classes,methods and interfaces**

Correct Answer:

d. **Classes,methods and interfaces**

Detailed Solution:

In **Java** (and many other object-oriented languages), a **package** is a **namespace** that organizes a set of **related classes, interfaces, and methods**.

QUESTION 6:

Consider the following 2 programs:

```
// Main1.java ------------------
public class Main1{
    public static void main(String args[]){
        int number = 10;
        System.out.println(number++ + ++number);
    }
}
```

```
// Main2.java ------------------
public class Main2{
    public static void main(String args[]){
        int number = 10;
        System.out.println(++number + number++);
    }
}
```

Choose the best option among the following for the code snippet given above

a. **Both pre-increment and post-increment operators becomes pre-increment during print.**

b. **Both pre-increment and post-increment operators becomes post-increment during print.**

c. **Both Main1 and Main2 classes give the same output.**

d. **Pre-increment and post-increment operators don't work during print.**

Correct Answer:

c. **Both Main1 and Main2 classes give the same output.**

Detailed Solution:

The output of both the program are 22. Therefore, option **c** is correct and we can eliminate option **d** that the operators don't work. Further, the operators are doing exactly what they are supposed to do i.e. pre-increment first increases the values and post-increment increases the value during the next operation. The print statement is the next operation; hence it received the post incremented value as well making option **a** and **b** invalid.

QUESTION 7:

Consider the following program:

```
class Base {
    public void print() {
        System.out.println("Base class...");
    }
}
class Derived extends Base {
    public void print() {
        System.out.println("Derived class...");
    }
}
public class Main {
    private static void main (String[] args) {
        Base b = new Base();
        b.print();
        Derived d = new Derived();
        d.print();
    }
}
```

How many errors does this program contain?

a. **None**

b. **1**

c. **2**

d. **3**

Correct Answer:

b. **1**

Detailed Solution:

This code has one error:
1. Incorrect visibility of `main` method:
The `main` method is defined as `private static void main(String[] args)`. The `main` method must be `public static void main(String[] args)` to be recognized as the entry point of the program by the Java Virtual Machine (JVM).

QUESTION 8:

Consider the code given below.

```
// Teacher.java ------------------
package nptel1;
public class Teacher {
  protected void showMarks() {
    System.out.println("100 Marks");
  }
}
```

```
// Student.java ------------------
package nptel2;
import nptel1.*;
public class Student extends Teacher {
  void show() {
    showMarks();
  }
  public static void main(String[] args) {
    Student st1 = new Student();
    st1.show();
  }
}
```

What is the output of the above Java Code Snippet with `protected` access modifier?

a. **100 marks**

b. **No output**

c. **Compiler error**

d. **None of the above**

Correct Answer:

a. **100 marks**

Detailed Solution:

Through inheritance, one can access a `protected` variable or method of a class even from outside the package. Here, we accessed `Teacher` class of `nptel1` from `Student` class of `nptel2`.

QUESTION 9:

Choose the correct syntax of a Java Package below.

a. **package PACKAGE_NAME;**

b. **package PACKAGE_NAME.*;**

c. **pkg PACKAGE_NAME;**

d. **pkg PACKAGE_NAME.*;**

Correct Answer:

a. **package PACKAGE_NAME;**

Detailed Solution:

A package declaration statement should end with a package name but not with *.

QUESTION 10:

What is the process by which we can control what parts of a program can access the members of a class?

a. **Polymorphism**

b. **Augmentation**

c. **Encapsulation**

d. **Recursion**

Correct Answer:

c. **Encapsulation**

Detailed Solution:

Encapsulation in Java is the process by which data (variables) and the code that acts upon them (methods) are integrated as a single unit. By encapsulating a class's variables, other classes cannot access them, and only the methods of the class can access them.