PROGRAMMING IN JAVA

Assignment 4

TYPE OF QUESTION: MCQ

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

QUESTION 1:

Which access modifier allows a method to be accessible within the same package but not from outside the package?

  a. **default**

  b. **private**

  c. **public**

  d. **protected**

Correct Answer:

  a. **default**

Detailed Solution:

The default access modifier (no explicit modifier specified) allows access to the method within the same package but not from outside. It is also known as "package-private."

QUESTION 2:

What will happen if you attempt to access a private method from another class in Java?

    a. **The method will be accessible.**

    b. **The method will throw an exception.**

    c. **The method will be inaccessible.**

    d. **The method will be converted to protected.**

Correct Answer:

    c. **The method will be inaccessible**

Detailed Solution:

A private method is accessible only within the same class. Any attempt to access it from another class results in a compilation error.

QUESTION 3:

What will be the output of the following code?

```java
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();
    }
}
```

a. 1 Java 10
   0 Java 0

b. 1 Java
   0 Java

c. 10 Java
   0 Java

d. 1 java 1
   0 Java 0

Correct Answer:

b. 1 Java
   0 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:

Which of the following is a built-in Java package?

     a. java.util

     b. my.package.util

     c. default.package

     d. system.java

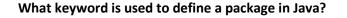**Correct Answer:**

     a. java.util

Detailed Solution:

`java.util` is a built-in Java package containing utility classes such as `ArrayList`, `HashMap`, and `Scanner`.

QUESTION 5:

What keyword is used to define a package in Java?

      a. define

      b. package

      c. import

      d. namespace

Correct Answer:

      b. package

Detailed Solution:

The `package` keyword is used in Java to define a package. For example, `package mypackage;` defines a package named `mypackage`.

QUESTION 6:

How do you import a specific class from a package in Java?

      a. **import package.*;**

      b. **import package.ClassName;**

      c. **include package.ClassName;**

      d. **use package.ClassName;**

Correct Answer:

      b. **import package.ClassName;**

Detailed Solution:

The syntax `import package.ClassName;` imports a specific class from a package. For example, `import java.util.Scanner;` imports the `Scanner` class from the `java.util` package.

QUESTION 7:

Consider the following program:

```java
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:

Which of the following is true about Java interfaces?

    a. **They cannot contain method implementations.**

    b. **An interface can extend multiple classes.**

    c. **An interface can only contain abstract methods.**

    d. **They allow a class to achieve multiple inheritance.**

Correct Answer:

    d. They allow a class to achieve multiple inheritance.

Detailed Solution:

Interfaces in Java provide a mechanism to achieve multiple inheritance because a class can implement multiple interfaces.

QUESTION 10:

Which access modifier provides the most restrictive access in Java?

      a. default

      b. protected

      c. private

      d. public

Correct Answer:

      c. private

Detailed Solution:

The `private` modifier provides the most restrictive access, allowing access only within the class where it is defined.