



---

## PROGRAMMING IN JAVA

### Assignment 4

TYPE OF QUESTION: MCQ

Number of questions: 15

Total mark:  $15 \times 1 = 15$

---

#### **QUESTION 1:**

For which purpose packages are used in Java?

- a. Categorizes data
- b. Organizing java classes into namespaces
- c. For faster compilation
- d. None

**Correct Answer:b**

#### **Detailed Solution:**

Java packages resolve the name conflicts among different classes.

---

#### **QUESTION 2:**

Which of the following keywords is used to define a package in Java?

- a. class
- b. implements
- c. extends
- d. package

**Correct Answer: d**

#### **Detailed Solution:**

For example: `package mypack;`

---

#### **QUESTION 3:**

Which of the following is an incorrect statement about packages?

- a. Package defines a namespace in which classes are stored.
- b. A package can contain other package within it.
- c. Java uses file system directories to store packages.



- 
- d. A package can be renamed without renaming the directory in which the classes are stored.

**Correct Answer: d**

**Detailed Solution:**

Yes, where you have defined the package (this is a .java file) only that need to be changed. No need to change the directory where a package is stored.

---

**QUESTION 4:**

Which of these access specifiers can be used for an interface?

- a. public
- b. private
- c. protected
- d. All of the above

**Correct Answer:a**

**Detailed Solution:**

Access specifier of an interface is either public or no specifier. When no access specifier is used then default access specifier is used due to which interface is available only to other members of the package in which it is declared, when declared public it can be used by any code.

---

**QUESTION 5:**

Which of the following is an incorrect statement about packages?

- a. Interfaces specifies what class must do but not how it does.
- b. Interfaces are specified public if they are to be accessed by any code in the program.
- c. All variables in interface are implicitly final and static.
- d. All variables are static and methods are public if interface is defined public.

**Correct Answer: d**

**Detailed Solution:**

This is according to the general rule of the interface.

---



---

### **QUESTION 6:**

Which one is correct declaration for implementing two interfaces?

- a. class C implements A, B { }
- b. class C implements A, implements B { }
- c. class C implements A extends B { }
- d. class C extend A, B { }

**Correct Answer:a**

#### **Detailed Solution:**

Here, (b) and (d) are syntactically wrong! (c) is valid but B is a class not an interface. Hence, the correct answer is (a)

---

### **QUESTION 7:**

The fields in an interface are implicitly specified as

- a. public
- b. protected
- c. private
- d. static and final

**Correct Answer: d**

#### **Detailed Solution:**

Fields in an interface are by default, static and final.

---

### **QUESTION 8:**

Let us consider the following piece of code in Java.

```
interface A {
    inti = 111;
}

class B implements A {
    void methodB() {
        i = 222;
        System.out.println(i);
    }
}
```

---

What will be the result of this code will be?

- a. There is no main () method so the program is not executable.
- b. The value of i will be printed as 111, as it is static and final by default.
- c. The value of i will be printed as 222, as it is initialized in class B.
- d. Compile time error

**Correct Answer: d**

**Detailed Solution:**

Because interface fields are static and final by default and you can't change their value once they are initialized. In the above code, methodB() is changing value of interface field A.i. It shows compile time error.

---

**QUESTION 9:**

Which of the following is/are true?

- 1) Every class is a part of some package.
  - 2) All classes in a file are part of the same package.
  - 3) If no package is specified, the classes in the file go into a special unnamed package.
  - 4) If no package is specified, a new package is created with folder name of class and the class is put in this package.
- 
- a. Only 1, 2 and 3
  - b. Only 3 and 4
  - c. Only 1 and 3
  - d. Only 3

**Correct Answer: a**

**Detailed Solution:**

This according to the property of package concept in Java.

---

**QUESTION 10:**

Which of the following class inheritance is/ are not supported in Java?

- a. Single inheritance
- b. Hybrid inheritance
- c. Multilevel inheritance
- d. Multiple inheritance

**Correct Answer: b, d**

**Detailed Solution:**

Hint: Hybrid inheritance and multiple inheritance from two classes are not possible.

---

**QUESTION 11:**

If a class inheriting an abstract class does not define all of its methods, then it will be known as

- a. Abstract class.
- b. A normal class.
- c. Final class
- d. An interface

**Correct Answer: a**

It cannot be a normal class as its method(s) remain(s) abstract (by virtue of inheritance). A class with final specifier is “final class” only. Further, a class with abstract method is not an interface; hence, (d) is also not true.

---

**QUESTION 12:**

Does a subclass inherit both member variables and methods?

- a. No—only member variables are inherited.
- b. No—only methods are inherited.
- c. Yes—both are inherited - but not those are declared as private.
- d. Yes—only the members/ methods with protected are inherited.

**Correct Answer: c**

**Detailed Solution:**

Both members and methods are inheritable except those are not declared as private.

---



---

**QUESTION 13:**

Can an object subclass another object?

- a. Yes—as long as single inheritance is followed.
- b. No—inheritance is only between classes.
- c. Only when one has been defined in terms of the other.
- d. Yes—when one object is used in the constructor of another.

**Correct Answer: b**

**Detailed Solution:**

Object is an instance of class; all inheritance are applicable to class and interface type only.

---

**QUESTION 14:**

What is the output for the following Java program?

```
class Base {
    public void show() {
        System.out.println("Base show() called");
    }
}

class Derived extends Base {
    public void show() {
        System.out.println("Derived show() called");
    }
}

public class Main {
    public void show() {
        System.out.println("Main show() called");
    }
    public static void main(String[] args) {
        Base bb = new Derived();
        bb.show();
    }
}
```

- a. Base show() called
- b. Main show() called
- c. Derived show() called
- d. Compile time error



---

**Correct Answer:c**

**Detailed Solution:**

As per the run-time polymorphism, *bb.show()* is bind to the *show()* of class Derived.

---

**QUESTION 15:**

Which of the following accessspecifiers can be used for a class so that its members can be accessed by a different class in the same package?

- a. public
- b. protected
- c. private
- d. default

**Correct Answer:d**

**Detailed Solution:**

A class with default access specifier is accessible to any class in the same file or any other class defined in another file but all of them belong to the same directory.

---

\*\*\*\*\*END\*\*\*\*\*