NPTEL Online Certification Courses

Indian Institute of Technology Kharagpur

NOC25-CS57 (JAN-2025 25S)

PROGRAMMING IN JAVA

Assignment 02

TYPE OF QUESTION: MCQ

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

QUESTION 1:

Consider the following object declaration statement

```
Scanner in = new Scanner(System.in);
```

What does `System.in` stands for in the above declaration?

a. **Any file storing data**

b. **Refers to the *standard input stream*, which is typically the keyboard by default.**

c. **Reference to scanner as an input device**

d. **It is a mouse as an input device**

Correct Answer:

b. **Refers to the *standard input stream*, which is typically the keyboard by default.**

Detailed Solution:

System.in refers to the standard input device and the keyboard is treated as standard input device. Thus, the code implies reading a data from keyboard.
Please refer to chapter 3 of book Joy With Java for a more detailed explaination.

QUESTION 2:

What will be the output of the following Java program?

```java
public class VarPrint {
    int x = 30;
    static int y = 20;

    public static void main(String[] args) {
        VarPrint t1 = new VarPrint();
        t1.x = 88;
        t1.y = 99;
        int z1 = t1.x + t1.y;
        VarPrint t2 = new VarPrint();
        System.out.println(t2.x + " " + t2.y + " " + z1);
    }
}
```

a. **30 99 178**

b. **30 88 129**

c. **30 99 187**

d. **88 99 178**

Correct Answer:

c. **30 99 187**

Detailed Solution:

If you perform any change for instance variable these changes won't be reflected for the remaining objects. Because for every object a separate copy of instance variable will be there. But if you do any change to the static variable, that change will be reflected for all objects because a static instance maintains a single copy in memory.
Please refer to chapter 3 of book Joy With Java for a more detailed explaination.

QUESTION 3:

What will be the output of the following Java program?

```java
public class ArgumenTest {
    public static void main(String[] args) {
        Test t = new Test();
        t.start();
    }

    static class Test {
        void start() {
            int a = 4;
            int b = 5;
            System.out.print("" + 8 + 3 + "");
            System.out.print(a + b);
            System.out.print(" " + a + b + "");
            System.out.print(foo() + a + b + " ");
            System.out.println(a + b + foo());
        }

        String foo() {
            return "foo";
        }
    }
}
```

a. **9 7 7 foo34 34foo**

b. **839 45foo45 9foo**

c. **72 34 34 foo34 34foo**

d. **9 7 7 foo 7 7foo**

Correct Answer:

b. **839 45foo45 9foo**

Detailed Solution:

Here, print() methods internally converts the data in its argument into a String object and then print the composition. Here, + is the concatenation of different String representation.
Please refer to chapter 3 of book Joy With Java for a more detailed explaination.

QUESTION 4:

What is encapsulation in object-oriented programming?

a. **Hiding implementation details and exposing only functionality**

b. **The process of creating multiple objects in a program**

c. **Writing multiple methods in a single class**

d. **Using the this keyword to reference an object**

Correct Answer:

a. **Hiding implementation details and exposing only functionality**

Detailed Solution:

Please refer to chapter 3 of book Joy With Java for a more detailed explaination.
Encapsulation is one of the fundamental principles of object-oriented programming. It involves bundling the data (variables) and the methods (functions) that operate on the data into a single unit, typically a class. By making certain data private and providing public methods to access or modify it, encapsulation helps hide implementation details from the outside world while exposing only the required functionality. This improves code modularity, security, and maintainability.

QUESTION 5:

Which of the following is true about constructors in a class?

a. **Constructors must have a return type.**

b. **Constructors are used to initialize objects.**

c. **A class can have only one constructor.**

d. **Constructors cannot be overloaded.**

Correct Answer:

b. **Constructors are used to initialize objects.**

Detailed Solution:

A constructor is a special method in a class that is automatically called when an object of the class is created. Its main purpose is to initialize the object's properties (variables). Unlike other methods, constructors:

- Have the same name as the class.
- Do not have a return type, not even `void`. A class can have multiple constructors with different parameter lists (constructor overloading) to allow flexibility in object creation.

Please refer book Joy with Java Chapter 3 for mored etailed explaination.

QUESTION 6:

What does the `this` keyword in Java help to achieve?

a. **Avoiding name space collision between instance variables and method parameters**

b. **Overloading methods in a class**

c. **Accessing private methods in another class**

d. **Creating multiple objects in a program**

Correct Answer:

a. **Avoiding name space collision between instance variables and method parameters**

Detailed Solution:

The *this* keyword is used to refer to the current instance of a class. It is commonly used to resolve ambiguity when instance variables and method parameters have the same name. For example:

```
class Example {
    int value;

    Example(int value) {
        this.value = value; // Resolves name collision by referring to
the instance variable
    }
}
```

Here, this.value refers to the instance variable, while value refers to the parameter of the constructor. This avoids confusion and ensures the proper assignment of values.
Please refer book Joy with Java Chapter 3 for mored etailed explaination.

QUESTION 7:

What is the correct signature of the `main` method in Java?

a. **public void main(String args[])**

b. **public static void main(String args[])**

c. **void main(String[] args)**

d. **public static void main(String[] args)**

Correct Answer:

d. **public static void main(String[] args)**

Detailed Solution:

The `main` method in Java must be declared as `public static void main(String[] args)` to be recognized by the JVM as the entry point of the program. The `public` modifier allows the method to be accessible from anywhere, `static` ensures it can be called without creating an instance of the class, and `String[] args` is the parameter used for command-line arguments.
Note: Please refer book Joy with Java Chapter 3 for mored etailed explaination.

QUESTION 8:

Which of the following is used to take runtime input in Java?

a. **BufferedReader**

b. **Scanner**

c. **DataInputStream**

d. **All of the above**

Correct Answer:

d. **All of the above**

Detailed Solution:

Java provides several ways to take runtime input:
`BufferedReader`: Reads text from the input stream efficiently.
`Scanner`: Parses primitive types and strings using regular expressions.
`DataInputStream`: Reads primitive data types and strings in a binary format. Each serves different purposes based on requirements.
Note: Please refer book Joy with Java Chapter 3 for mored etailed explaination.

QUESTION 9:

Which method is used to format output in Java?

a. **print()**

b. **println()**

c. **printf()**

d. **format()**

Correct Answer:

c. **printf()**

Detailed Solution:

The `printf()` method is used in Java to produce formatted output. It follows the syntax of `System.out.printf(format, arguments)` where the format specifies how the output should appear. For example:

```
System.out.printf("%d %s", 5, "Apples");
```

prints "5 Apples".

Note: Please refer book Joy with Java Chapter 3 for mored etailed explaination.

QUESTION 10:

Which Java class is primarily used to read input from the console?

a. **Scanner**

b. **BufferedReader**

c. **Console**

d. **DataInputStream**

Correct Answer:

a. **Scanner**

Detailed Solution:

The `Scanner` class is widely used to read input from the console in Java. It provides methods to parse and read primitive types (e.g., `nextInt()`, `nextDouble()`) and strings (e.g., `next()`, `nextLine()`). It is a versatile and easy-to-use class for input handling.
Note: Please refer to chapter 3 of book Joy With Java for a more detailed explaination.