PROGRAMMING IN JAVA

Assignment 2

TYPE OF QUESTION: MCQ

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

QUESTION 1:

Which of the following is the correct way to declare a class in Java?

a. public class MyClass {}

b. class MyClass[] {}

c. public MyClass class {}

d. MyClass public class {}

Correct Answer:

a. public class MyClass {}

Detailed Solution:

The correct way to declare a class in Java is by using the `class` keyword followed by the class name and curly braces. Refer to Lecture 7 for more details.

QUESTION 2:

What is the purpose of a constructor in a class?

a. **To destroy objects of the class**

b. **To create static methods**

c. **To implement inheritance**

d. **To initialize objects of the class**

Correct Answer:

d. **To initialize objects of the class**

Detailed Solution:

A constructor is used to initialize objects of a class. It is called when an instance of the class is created and can set initial values for object attributes. Refer to Lecture 7 for more details.

QUESTION 3:

Which keyword is used in Java to refer to the current object?

a. **that**

b. **self**

c. **current**

d. **this**

Correct Answer:

d. **this**

Detailed Solution:

In Java, the `this` keyword is used to refer to the current object within an instance method or a constructor. Refer to Lecture 8 for more details.

QUESTION 4:

Consider the following code snippet. What will be the output?

```
class NPTEL_W2 {
  int x;

  NPTEL_W2(int x) {
      this.x = x;
  }

  void printX() {
      System.out.println(this.x);
  }

  public static void main(String[] args) {
      NPTEL_W2 obj = new NPTEL_W2(10);
      obj.printX();
  }
}
```

a. 0

b. 10

c. Compilation error

d. Runtime error

Correct Answer:

b. 10

Detailed Solution:

The constructor `NPTEL_W2 (int x)` initializes the instance variable `x` with the value passed as an argument. The method `printX()` prints the value of `x`, which is `10`. Refer to Lecture 7 for more details.

QUESTION 5:

Which of the following demonstrates constructor overloading in Java?

- **a. Defining multiple constructors in a class with different parameter lists**
- **b. Defining multiple methods in a class with the same name**
- **c. Defining a constructor in a subclass**
- **d. Using the super keyword**

Correct Answer:

- **a. Defining multiple constructors in a class with different parameter lists**

Detailed Solution:

Constructor overloading in Java is achieved by defining multiple constructors in a class, each with different parameter lists. This allows creating objects in different ways. Refer to Lecture 8 for more details.

QUESTION 6:

What is the purpose of the this keyword in the context of avoiding name space collision?

a. **To call another constructor in the same class**

b. **To refer to the current object**

c. **To differentiate between instance variables and parameters with the same name**

d. **To import another class**

Correct Answer:

c. **To differentiate between instance variables and parameters with the same name**

Detailed Solution:

The this keyword is used to differentiate between instance variables and parameters when they have the same name, avoiding name space collision. Refer to Lecture 8 for more details.

QUESTION 7:

Which of the following is the correct signature of the main method in Java?

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

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

c. **public static void main()**

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

Correct Answer:

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

Detailed Solution:

The correct signature of the `main` method in Java is `public static void main(String[] args)`. This method serves as the entry point for the Java application. Refer to Lecture 9 for more details.

QUESTION 8:

Which class is used in Java to take runtime data input from the user?

- **a. BufferReader**
- **b. UserInputStreamReader**
- **c. Scanner**
- **d. DataInputStreamReader**

Correct Answer:

c. Scanner

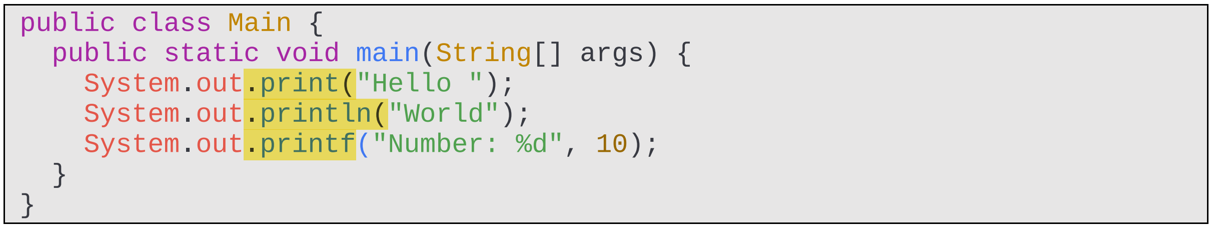
Detailed Solution:

The Scanner class is used to take runtime data input from the user. It provides methods to read various types of input such as strings, integers, and floating-point numbers. Refer to Lecture 9 for more details.

```
import java.util.Scanner;
public class Main {
  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Enter a string: ");
    String str = scanner.nextLine();
    System.out.print("Enter an integer: ");
    int integer = scanner.nextInt();
    System.out.print("Enter a double: ");
    double dbl = scanner.nextDouble();
    System.out.print("Enter a boolean (true/false): ");
    boolean bool = scanner.nextBoolean();
    System.out.println("You entered:");
    System.out.println("String: " + str);
    System.out.println("Integer: " + integer);
    System.out.println("Double: " + dbl);
    System.out.println("Boolean: " + bool);
    scanner.close();
  }
}
```

QUESTION 9:

What is the output of the following Java code snippet?
(\n in output is to be assumed to be the new line character)



```
public class Main {
  public static void main(String[] args) {
    System.out.print("Hello ");
    System.out.println("World");
    System.out.printf("Number: %d", 10);
  }
}
```

a. **Hello World\nNumber: 10**

b. **Hello WorldNumber: 10**

c. **Hello \nWorld\nNumber: 10**

d. **Hello World\nNumber: 10\n**

does printf give new line char at the end?

No, printf in Java does not automatically add a newline character at the end. If you want a newline, you need to include it explicitly with \n (or %n for platform-independent newline).

Correct Answer:

a. **Hello World\nNumber: 10**

Detailed Solution:

The `print` method prints text without a newline, `println` prints text with a newline, and `printf` prints formatted text. The output is `Hello World` on the first line and `Number: 10` on the second line. Refer to Lecture 10 for more details.

QUESTION 10:

How do you read a line of text from the console using the Scanner class in Java?

a. **scanner.readLine()**

b. **scanner.nextLine()**

c. **scanner.getLine()**

d. **scanner.fetchLine()**

Correct Answer:

b. **scanner.nextLine()**

Detailed Solution:

The nextLine() method of the Scanner class reads a line of text from the console. Refer to Lecture 10 for more details.