NPTEL Online Certification Courses
Indian Institute of Technology Kharagpur
NOC24-CS105 (July-2024 24A)

PROGRAMMING IN JAVA
Assignment 06
TYPE OF QUESTION: MCQ

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

QUESTION 1:

What is the output of the following program?

```java
public class Question {
  public static void main(String[] args) {
    try {
      int a = 5 / 0;
    }
    catch (Exception e) {

        catch (ArithmeticException a) {
          System.out.println("Cannot divide by 0");
        }
    }
    System.out.println("Hello World");
  }
}
```

a. "Hello World"

b. "Cannot divide by 0"

c. Compilation Error

d. Runtime Error (the code compiles successfully)

Correct Answer:

c. Compilation Error

Detailed Solution:

This first handler catches exceptions of type `Exception`; therefore, it catches any exception, including `ArithmeticException`. The second handler could never be reached. This code will not compile.

QUESTION 2:

What will be the output of the following program?

```
class Question {
    int i; 21
    public Question(int i) {   21
      21 this.i = i--; 20
    }
}
class Question1 extends Question {
    public Question1(int i) {20
        super(++i); 21
        System.out.println(i);
    }
}
public class Check {
    public static void main(String[] args) {
        Question1 n = new Question1(20);
    }
}
```

a. **20**

[✓ **b. 21**]

c. **19**

d. **22**

Correct Answer:

b. 21

Detailed Solution:

The program creates an instance of `Question1` with an initial value of `20`. Inside the `Question1` constructor, `i` is pre-incremented to `21` before being passed to the `Question` superclass constructor. In the `Question` constructor, `this.i` is assigned the value `21` (from `i--` which uses `i` before decrementing it). Returning to the `Question1` constructor, `i` is still `21`, and this value is printed. Thus, the output is `21`.

QUESTION 3:

What is the output of the following program?

```
class Question {
    static int x;
    static {
        x++;
    }

    {
        ++x;
    }
}
class Question1 extends Question {
    static {
        --x;
    }

    {
        x--;
    }
}
public class Check {
    public static void main(String[] args) {
        System.out.println(new Question1().x);
    }
}
```

a. 1

b. 2

c. 0

d. Compilation Error

Key Points
Instance Initialization Block: Useful for common initialization tasks shared by all constructors.
Constructor: Primarily for initializing instance variables and can accept parameters for flexibility.
Both are used for initializing object state but serve slightly different purposes in terms of functionality and design.

Correct Answer:

c. 0

Detailed Solution:

When the `Question` class is loaded, its static block increments `x` to `1`. Then, the `Question1` class is loaded, and its static block decrements `x` back to `0`. When an instance of `Question1` is created, the instance initializer blocks run: `Question`'s block increments `x` to `1`, and `Question1'`s block decrements `x` back to `0`. Therefore, printing `x` in the `main` method outputs `0`.

QUESTION 4:

Which exception is thrown when an array element is accessed beyond the array size?

a. **ArrayElementOutOfBounds**

b. **ArrayIndexOutOfBoundsException**

c. **ArrayIndexOutOfBounds**

d. **None of these**

Correct Answer:

b. **ArrayIndexOutOfBoundsException**

Detailed Solution:

The exception that is thrown when an array element is accessed beyond the array size in Java is:

```
ArrayIndexOutOfBoundsException
```

This is the specific exception in Java that indicates that an array has been accessed with an illegal index, either negative or greater than or equal to the size of the array. The other options listed do not represent valid exceptions in Java.

QUESTION 5:

What is the output of the following program?

```
class Q {
    public void disp() {
        System.out.println("java");
    }
}
class P extends Q {
    public void disp() {
        System.out.println("nptel");
    }
}
class C extends P {
    public void disp(){
        super.super.disp();
        System.out.println("course");
    }
}
public class Question {
    public static void main(String[] args) {
        C c = new C();
        c.disp();
    }
}
```

a. **java**

b. **java**
   course

c. **nptel**
   course

d. **Compilation Error**

Correct Answer:

d. **Compilation Error**

Detailed Solution:

The code attempts to use `super.super.disp()` in the `C` class, which is invalid in Java. Java does not support accessing a grandparent class's method directly using `super.super`. This syntax results in a compilation error because `super` can only be used to refer to the immediate superclass.

QUESTION 6:

Fill in the blank in the program so that the output is “Java”.

```
interface X {
    void display();
}
class Y implements X {
    ___________ display() {              //MISSING_CODE
        System.out.println("Java");
    }
}
public class MainClass {
    public static void main(String[] args) {
        Y r = new Y();
        r.display();
    }
}
```

a. **public void**

b. **void**

c. **private void**

d. **static void**

Correct Answer:

a. **public void**

Detailed Solution: abstract+PUBLIC

Interface methods must be implemented as `public`. Because, interface methods are `public` by default and you should not reduce the visibility of any methods while overriding.

QUESTION 7:

How many times will "Java" be printed if the following code is executed?

```
class X {
    static {
        Y.display();
    }
}
class Y extends X {
    static void display() {
        System.out.println("Java");
    }
}
public class MainClass {
    public static void main(String[] args) {
        Y.display();
    }
}
```

a. 0

b. 1

c. 2

d. 3

Correct Answer:

c. 2

Detailed Solution:

When the code is executed, the static block in class X is executed first, which calls `Y.display()`. This prints "Java" once. Then, in the `main` method of `MainClass`, `Y.display()` is called again, resulting in "Java" being printed for the second time. Thus, "Java" is printed twice in total.

QUESTION 8:

The following is a simple program using the concept of thread. What is the output of the following program?

```
public class Question extends Thread {
  public void run() {
    for (int i = 1; i < 8; i++) {
      System.out.print(++i + " ");
    }
  }
  public static void main(String args[]) {
    Question t1 = new Question();
    t1.run();
  }
}
```

a. **1 3 5 7**

b. **2 4 6 8**

c. **1 3 5 7 9**

d. **2 4 6**

Correct Answer:

b. **2 4 6 8**

Detailed Solution:

The increment operators increase the value of `i` to 2 in the first run. Afterwards, two increments are happening till `i < 8` condition is not satisfied.

QUESTION 9:

For the program given below, what will be the output after its execution?

```
public class Main {
  public static void main(String[] args) {
    Thread thread = Thread.currentThread();
    thread.run();
    System.out.print(Thread.activeCount());
  }
}
```

a. 1

b. 2

c. 0

d. 01

Correct Answer:

a. 1

Detailed Solution:

`java.lang.Thread.activeCount()` : Returns an estimate of the number of active threads in the current thread's thread group and its subgroups.

QUESTION 10:

Which of the following method returns a reference to the currently executing thread object?

a. **public static boolean interrupted();**

b. **public static Thread currentThread();**

c. **public final boolean isAlive();**

d. **public final void suspend();**

Correct Answer:

b. **public static Thread currentThread();**

Detailed Solution:

Only `public static Thread currentThread()` method returns a reference to the currently executing thread object among the options.