

## Java MCQs - 14

Topics : [JAVA](#)

Written on [March 19, 2024](#)

161. Consider the following code:

```
public class Main {  
    public static void main(String[] args) {  
        int[] arr = {1, 2, 3, 4, 5};  
        System.out.println(arr[5]);  
    }  
}
```

What is the output of the above code?

- A) 1
- B) 2
- C) 3
- D) `ArrayIndexOutOfBoundsException`

**Answer: D) `ArrayIndexOutOfBoundsException`**

Explanation: The array `arr` has indices from 0 to 4, and accessing index 5 will result in an `ArrayIndexOutOfBoundsException`.

162. Which of the following statements about Java generics is true?

- A) Generics are a way to achieve multiple inheritance in Java.
- B) Generics are used to enforce type safety at compile-time.
- C) Generics can only be used with primitive data types.
- D) Generics can only be used with classes and not interfaces.

**Answer: B) Generics are used to enforce type safety at compile-time.**

163. Consider the following code:

```
public class Main {  
    public static void main(String[] args) {  
        int x = 5;  
        System.out.println(x++ + ++x);  
    }  
}
```

What is the output of the above code?

- A) 11
- B) 12
- C) 13
- D) Compilation Error

**Answer: C) 13**

Explanation: `x++` increments `x` after the current value is used in the expression, while `++x` increments `x` before the current value is used in the expression.

164. Which of the following statements about Java threads is true?

- A) A thread in Java is always a lightweight process.
- B) Java threads always have the same priority.
- C) Threads cannot be created in Java.
- D) Threads share the same memory space.

**Answer: A) A thread in Java is always a lightweight process.**

165. Consider the following code:

```
public class Main {  
    public static void main(String[] args) {  
        String str1 = "Hello";  
        String str2 = new String("Hello");  
        System.out.println(str1 == str2);  
    }  
}
```

What is the output of the above code?

- A) true
- B) false
- C) Compilation Error
- D) Runtime Error

**Answer: B) false**

Explanation: `str1` and `str2` are two different objects in memory. The `==` operator checks for reference equality, and since they are different objects, the result is `false`.

166. Which of the following statements about Java exceptions is true?

- A) All exceptions in Java are checked exceptions.
- B) Checked exceptions are subclasses of `RuntimeException`.
- C) Unchecked exceptions must be caught using a `try-catch` block.
- D) The `throws` keyword is used to handle exceptions.

**Answer: B) Checked exceptions are subclasses of RuntimeException.**

167. Consider the following code:

```
public class Main {
    public static void main(String[] args) {
        int x = 5;
        int y = 0;
        try {
            int z = x / y;
            System.out.println(z);
        } catch (ArithmeticException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

What is the output of the above code?

- A) 0
- B) Error: / by zero
- C) Runtime Error
- D) Compilation Error

**Answer: B) Error: / by zero**

Explanation: An attempt to divide by zero results in an `ArithmeticException`, which is caught by the catch block.

168. What is the purpose of the `break` statement in Java?

- A) To terminate the execution of a loop or switch statement.
- B) To skip the current iteration of a loop and continue with the next iteration.
- C) To define a default case in a switch statement.
- D) To exit the entire program.

**Answer: A) To terminate the execution of a loop or switch statement.**

169. Consider the following code:

```
public class Main {
    public static void main(String[] args) {
        int[] arr = new int[5];
        System.out.println(arr.length);
    }
}
```

What is the output of the above code?

- A) 0
- B) 1
- C) 5
- D) Compilation Error

**Answer: C) 5**

Explanation: `arr.length` returns the length of the array, which is 5.

170. Which of the following statements about Java I/O streams is true?

- A) Java I/O streams are only used for network communication.
- B) Java I/O streams can be classified into three types: byte streams, character streams, and object streams.
- C) Java I/O streams support only character-based input/output.
- D) Java I/O streams are used only for reading data and not for writing.

**Answer: B) Java I/O streams can be classified into three types: byte streams, character streams, and object streams.**

© Copyright **Aryatechno**. All Rights Reserved. Written tutorials and materials by [Aryatechno](#)

ARYATECHNO