Task on 16/10/2025

Topic: Exception Handling in Java (try/catch/finally, try-with-resources, multi-catch, custom exceptions)

Question 1

Given:

```
try {
    int x = 10 / 0;
    System.out.println("Result: " + x);
} catch (ArithmeticException e) {
    System.out.println("Divide by zero");
}
```

What is printed?
A. Result: 0
B. Divide by zero
C. Compilation error
D. Runtime error

Question 2

Given:

```
try {
    int[] nums = {1, 2, 3};
    System.out.println(nums[3]);
} catch (ArrayIndexOutOfBoundsException e) {
    System.out.println("Index out of bounds");
} finally {
    System.out.println("Finally block executed");
}
```

What is the output?
A. Index out of bounds
B. Finally block executed
C. Both A and B
D. Runtime error

Question 3

Given:

```
try {
    int a = 10 / 2;
} finally {
    System.out.println("Finally runs");
}
```

What happens?
A. Compilation error
B. Runtime error
C. Prints "Finally runs"
D. Nothing is printed

Question 4

Which of the following statements is **true** about the `finally` block?
A. It executes only when an exception occurs.
B. It executes only when no exception occurs.
C. It always executes, regardless of whether an exception occurs or not.
D. It executes only when explicitly called.

Question 5

Given:

```
try {
    String s = null;
    System.out.println(s.length());
} catch (NullPointerException e) {
    System.out.println("Null value");
} catch (Exception e) {
    System.out.println("General exception");
}
```

What is printed?
A. Null value
B. General exception
C. Compilation error
D. Runtime error

Question 6

Given:

```
try {
    int a = Integer.parseInt("ABC");
} catch (NumberFormatException | NullPointerException e) {
    System.out.println("Invalid number");
}
```

What is printed?
A. Compilation error
B. Invalid number
C. Runtime error
D. None

Question 7

What is the rule for multi-catch blocks?
A. The caught exceptions must have a parent-child relationship.
B. The caught exceptions must be unrelated.
C. Only one exception type can be caught.
D. Checked exceptions cannot be used.

Question 8

Given:

```
try (java.io.FileReader fr = new
java.io.FileReader("file.txt")) {
    System.out.println("Reading file");
} catch (java.io.IOException e) {
    System.out.println("IO error");
}
```

What feature is used here?
A. finally block
B. multi-catch block
C. try-with-resources
D. nested try-catch

Question 9

Why is the **try-with-resources** statement preferred?
A. It suppresses all exceptions.
B. It automatically closes resources.
C. It prevents checked exceptions.
D. It skips the catch block.

Question 10

Given:

```
class MyException extends Exception {
    MyException(String msg) { super(msg); }
}

public class Test {
    public static void main(String[] args) {
        try {
            throw new MyException("Custom error occurred");
        } catch (MyException e) {
            System.out.println(e.getMessage());
        }
    }
}
```

What is printed?
A. Nothing
B. Custom error occurred
C. Exception
D. Compilation error

Question 11

Which statement about custom exceptions is true?
A. They must extend `Throwable` directly.
B. They must extend either `Exception` or `RuntimeException`.
C. They cannot include constructors.
D. They cannot be thrown explicitly.

Question 12

Given:

```
try {
    throw new RuntimeException("Test");
} catch (RuntimeException e) {
    throw e;
} finally {
    System.out.println("Finally executed");
}
```

What is the output?
A. Test
B. Finally executed
C. Finally executed, then RuntimeException
D. Compilation error

Question 13

Given:

```
try {
    System.out.println("Start");
    System.exit(0);
} finally {
    System.out.println("End");
}
```

What happens?
A. Prints “Start” then “End”
B. Only prints “Start”
C. Compilation error
D. Prints “End” only

Question 14

Given:

```
try {
    int a = 5 / 0;
} catch (ArithmeticException e) {
    throw new IllegalArgumentException("Illegal Argument");
}
```

What happens?
A. Catches ArithmeticException and rethrows as IllegalArgumentException
B. Both exceptions are printed
C. Program terminates silently
D. Compilation error

Question 15

Given:

```
class CustomException extends Exception {
    public CustomException(String msg) { super(msg); }
}

public class Demo {
    public static void main(String[] args) throws
CustomException {
        try {
            throw new CustomException("Demo error");
        } finally {
            System.out.println("Finally block");
        }
    }
}
```

What happens when executed?
A. Prints “Finally block” then terminates with CustomException
B. Only prints “Finally block”
C. Prints “Demo error”
D. Compilation error
