

1. Explain different types of Errors in Java

Syntax Errors: Syntax errors occur during compilation due to violations of Java's syntax rules.

Runtime Errors: Occur during execution, such as accessing an array index out of bounds or dividing by zero.

Logical Errors: Errors in the logic of the program that lead to incorrect output.

JVM Errors: These include errors like `StackOverflowError` or `OutOfMemoryError` that occur due to resource limitations or JVM malfunction.

2. What is an Exception in Java

An exception is an event that disrupts the normal flow of a program's execution. It is an object that Java uses to signal that an error or unusual situation has occurred. Exceptions are instances of the `Throwable` class or its subclasses

3. How can you handle exceptions in Java? Explain with an example

Java provides the try-catch-finally mechanism to handle exceptions gracefully and ensure that the program continues to execute even after encountering errors.

```
try {
    int result = 10 / 0; // This will throw ArithmeticException
    System.out.println(result);
}
catch (ArithmeticException e) {
    System.out.println("Exception caught: Division by zero is not allowed.");
}
finally {
    System.out.println("This block always executes.");
}
```

4. Why do we need exception handling in Java?

- Exception handling allows the program to continue running or terminate gracefully even after encountering unexpected errors.
- Without exception handling, runtime errors can terminate the program. Exception handling ensures controlled error resolution without the termination of program.
- Exception handling ensures that the program behaves predictably even when encountering unexpected situations.
- By catching exceptions, we developers can log errors, making it easier to debug and identify issues in the application.

5. What is the difference between exception and error in Java?

Exceptions :

- Problems that can be caught and handled.
- Subclass of Throwable.
- Ex - IOException, ArithmeticException, etc

Errors :

- Serious issues that cannot be recovered from.
- Ex - OutOfMemoryError, StackOverflowError

6. Name the different types of exceptions in Java

Checked Exceptions:

Exceptions checked at compile time. Examples:

- IOException
- SQLException

Unchecked Exceptions:

Exceptions that occur at runtime and are not checked at compile time. Examples:

- ArithmeticException
- NullPointerException
- ArrayIndexOutOfBoundsException

Errors:

Serious issues related to the JVM that are not meant to be handled by the program. Examples:

- OutOfMemoryError
- StackOverflowError

7. Can we just use try instead of finally and catch blocks?

A try block must always be accompanied by either a catch block, a finally block, or both.

Example :

```
try {
    int x = 5/0; // exception
    System.out.println(x);
}
catch (ArithmeticException e) {
    System.out.println("Exception caught: Division by zero is not allowed.");
}
finally {
    System.out.println("This block always executes.");
}
```