

### 1. Identify Exception Causes:

- List five common reasons why exceptions might occur in a Java program.
- Explain the difference between `Error` and `Exception`.

### 2. Working with Built-in Exceptions:

- Write a code snippet that demonstrates the handling of an `ArithmeticException` using a `try-catch` block.
- Modify the code to include a `finally` block for resource cleanup.

### 3. Custom Exceptions:

- Create a custom exception class called `InvalidAgeException` that extends `Exception`.
- Write a method that checks a user's age and throws `InvalidAgeException` if the age is less than 18.

### 4. Exception Details:

- Use the `printStackTrace()`, `toString()`, and `getMessage()` methods to print detailed information about an exception in a program.
- Compare and explain the output of these methods.

### 5. Multiple Catch Blocks:

- Write a program that generates both an `ArrayIndexOutOfBoundsException` and a `NullPointerException`.
- Use multiple `catch` blocks to handle each exception and print a custom message for each.