1. **Explain different types of Errors in Java**

In any programming language we categorize errors into 2 type

- Syntax Error/CompileTime Mistakec
- Logical Error/RunTimeMistakes

Syntax error/CompileTime MistakeE

- It refers to the mistakes done by the programmer with respect to syntax
- These mistakes are identified by the compiler, so we say it as“CompileTimeMistake”.

Logical Error/RunTimeMistake

- It refers to the mistakes done by the programmer in terms of writing a logic.
- These mistakes are identified by jvm during the execution of a program, so we say it as “Runtime Mistake”.

2. **What is an Exception in Java?**
An exception in Java is an event that disrupts the normal flow of the program's instructions. It is an object that is thrown at runtime when an error occurs. Exceptions can be caught and handled to prevent the program from crashing.

3. **How can you handle exceptions in Java? Explain with an example**
Exceptions in Java can be handled using try, catch, and finally blocks.
Here's an example:

```
try {
    int result = 10 / 0; // This will throw an ArithmeticException
}
catch (ArithmeticException e) {
    System.out.println("Cannot divide by zero: " + e.getMessage());
}
finally {
    System.out.println("This block is always executed.");
}
```

4. **Why do we need exception handling in Java?**
Exception handling is necessary to:

- ⊄ Maintain the normal flow of the application.
- ⊄ Handle runtime errors gracefully.
- ⊄ Provide a mechanism to recover from unexpected conditions.
- ⊄ Improve the robustness and reliability of the code.

5. **What is the difference between exception and error in Java?**
   - ⊄ **Exception**:
     - Represents conditions that a program might want to catch.
     - Can be checked or unchecked.
     - Examples: IOException, SQLException.
   - ⊄ **Error**:
     - Represents serious problems that a reasonable application should not try to catch.
     - Always unchecked.
     - Examples: OutOfMemoryError, StackOverflowError.
     -
6. **Name the different types of exceptions in Java.**
   - ⊄ **Checked Exceptions**: Must be either caught or declared in the method signature. Examples: IOException, SQLException.
   - ⊄ **Unchecked Exceptions**: Do not need to be declared or caught. Examples: NullPointerException, ArithmeticException.
   - ⊄
7. **Can we just use try instead of finally and catch blocks?**
   No, a try block must be followed by either a catch block or a finally block, or both. The catch block is used to handle exceptions, while the finally block is used to execute code that must run regardless of whether an exception was thrown or not.