**What is Exception?**

Exception is an abnormal condition.

**What is Error?**

Errors represent irrecoverable conditions.

we should not try to handle errors.

Example:

running out of memory, memory leaks, stack overflow errors, library incompatibility, infinite recursion, etc.

**What is Exception Handling?**

Exception Handling is a mechanism to handle runtime errors such as ClassNotFoundException, IOException, SQLException, RemoteException, etc.

**How many types of Exceptions in Java?**

There are two types of exceptions in java.

checked and unchecked exceptions.

**Explain checked exceptions?**

Checked exceptions are called compile-time exceptions because these exceptions are checked at compile-time by the compiler.

Exception class is the superclass for all checked exceptions except IOException.

Example: IOException,SQLException, ParseException, ClassNotFoundException etc.

**Explain unchecked exceptions?**

Unchecked exceptions are called run-time exceptions because these exceptions will not check at compile-time by the compiler.

Exceptions under Error and RuntimeException classes are unchecked exceptions.

Example: ArithmeticException, ArrayIndexOutOfBoundsException, NullPointerException, NumberFormatException,

**How to handle Exceptions in java?**

The code that may throw an exception is enclosed within the try block, and the exception handling code is written in the catch block.

finally block is used to execute code that must run regardless of whether an exception is thrown or not.

Throw keyword is used to explicitly throw an exception.

Throws keyword is used in method signatures to declare that a method can throw certain exceptions.

In java, Throwable class is the root class of Java Exception and inherited by two subclasses Exception and Error.

**What is difference between Throw and Throws?**

Throw:

The throw keyword is used to explicitly throw an exception from a method or any block of code.

Throw is used within a method body.

Throw is followed by an instance of Throwable.

Unchecked exceptions do not need to be declared in the method signature those must be throw.

Throws:

The throws keyword is used in method signatures to declare that a method can throw one or more exceptions.

Throws is used in the method declaration.

Throws is followed by a comma-separated list of exception classes.

Checked exceptions must be declared in the method signature using throws.

**Explain about Throwable?**

Throwable is the superclass for all errors and exceptions that can be thrown by the Java Virtual Machine (JVM) or by application code.

Various methods like getMessage(), toString(), and printStackTrace() help in debugging and handling exceptions.

**Explain broad exception and narrow exception?**

Broad exception:

By catching a broad exception, you are handling multiple types of exceptions, including any subclass of the broad exception.

It is not specific to the actual error that occurred.

It reduce the number of catch blocks required.

Exception class is a broad exception class that catches any exception.

Narrow Exception:

A narrow exception refers to catching specific exception.

By catching a narrow exception, we handle a specific type of error.

It simplifies debugging by providing detailed handling.

**How to create custom exception in java?**

Create a class for custom exception.

For checked exceptions we extend Exception.

For unchecked exceptions we extend RuntimeException.

Provide constructors that call the corresponding constructors of the superclass.

For custom checked exceptions declared in the method signature using throws.

For custom unchecked exceptions do not need to declare it using the throws keyword.

**can we handle exceptions try without catch?**

Yes, we can handle exceptions using a try block without a catch block by using a finally block instead.

**what is try-with-resources?**

simplifies resource management by automatically closing resources when the try block exits to avoid resource leaks.

It is useful for dealing with resources such as files, database connections, sockets, etc.

To use try-with-resources with custom classes, your class must implement the AutoCloseable interface.

You can manage multiple resources in a single try-with-resources statement by separating them with semicolons.

**How to handle exceptions in spring boot?**

The @ExceptionHandler annotation is used within a controller class to handle specific exceptions thrown by request handler methods.

The @ControllerAdvice annotation is used to handle exceptions globally across all controllers.

The GlobalExceptionHandler class will catch exceptions thrown by any controller in the application, allowing for centralized exception handling.

The ResponseStatusException is a runtime exception that can be thrown to return a specific HTTP status code and message.

The @ResponseStatus annotation can be used on custom exceptions to specify the HTTP status code and reason to be returned.

@RestControllerAdvice is a specialization of @ControllerAdvice that applies to RESTful APIs, combining @ControllerAdvice and @ResponseBody.

**can we throws for main method?**

Yes, we can declare the main method to throw exceptions in Java.