

Q1.What is Exception in Java?

Ans

An exception is an event that occurs during the execution of a program that disrupts the normal flow of instructions.

Q2.What is Exception Handling?

Ans

Exception handling is the process of responding to unwanted or unexpected events when a computer program runs.

Q3.What is the difference between Checked and Unchecked Exceptions and Error?

Ans

Checked Exception-

Checked exceptions occur at compile time.

The compiler checks a checked exception.

These types of exceptions can be handled at the time of compilation.

They are the sub-class of the exception class.

Unchecked Exception-

Unchecked exceptions occur at runtime.

The compiler does not check these types of exceptions.

These types of exceptions cannot be a catch or handle at the time of compilation, because they get generated by the mistakes in the program.

They are runtime exceptions and hence are not a part of the Exception class.

Q4.What are the difference between throw and throws in Java?

Ans

We use the throws keyword to declare what exceptions we can throw from a method. The throw keyword, on the other hand, is mainly used to throw an exception explicitly within a block of code or a method. We can use the throws keyword in a method signature. It declares what exceptions a method can throw.

Q5.What is multithreading in Java? mention its advantages

Ans

Multithreading is a Java feature that allows concurrent execution of two or more parts of a program for maximum utilization of CPU. Each part of such program is called a thread. So, threads are light-weight processes within a process.

Advantages-

- Enhanced performance by decreased development time
- Simplified and streamlined program coding
- Improved GUI responsiveness
- Simultaneous and parallelized occurrence of tasks
- Better use of cache storage by utilization of resources
- Decreased cost of maintenance
- Better use of CPU resource

Q6. Write a program to create and call a custom exception

Ans

```
class CustomException extends Exception {  
  
    public CustomException(String message) {  
        super(message);  
    }  
}  
  
public class Main {  
  
    public static void main(String[] args) {  
        try {  
            throw new CustomException("This is my custom exception.");  
        } catch (CustomException e) {  
            System.out.println(e.getMessage());  
        }  
    }  
}
```

Q7. How can you handle exceptions in Java?

Ans

There are two ways to handle exceptions in Java:

Try-catch-finally block: This is the most common way to handle exceptions in Java. It consists of three parts:

The try block: This block contains the code that may throw an exception.

The catch block: This block is used to handle the exception.

The finally block: This block is executed whether an exception is thrown or not. It is often used to close resources, such as files or sockets.

Try-with-resources block: This block is a newer way to handle exceptions in Java. It is used to simplify the code for handling resources that need to be closed.

Q8.What is Thread in Java?

Ans

A thread in Java is a lightweight process that allows a program to run multiple tasks simultaneously. Threads are independent of each other and can run concurrently. This means that multiple threads can be running at the same time, even if they are part of the same program.

Threads are created using the Thread class. This class provides methods for starting, stopping, and managing threads.

Q9. What are the two ways of implementing thread in Java?

Ans

There are two ways of implementing thread in Java:

Extend the Thread class: This is the traditional way to create a thread. To do this, you need to create a class that extends the Thread class. The run() method of the Thread class is overridden to provide the code that the thread will run.

Implement the Runnable interface: This is a newer way to create a thread. To do this, you need to create a class that implements the Runnable interface. The run() method of the Runnable interface is overridden to provide the code that the thread will run.

Q10.What do you mean by garbage collection?

Ans

Garbage collection is a process by which the Java Virtual Machine (JVM) automatically deallocates memory that is no longer being used by an application. This frees up memory for other objects that need to be created, and it prevents the application from running out of memory.

In Java, objects are created on the heap, which is a portion of memory that is dedicated to the program. When an object is no longer needed, it is considered garbage. The garbage collector periodically scans the heap for garbage objects, and it deallocates the memory that they are using.

Garbage collection is an automatic process, so the programmer does not need to explicitly deallocate memory. This makes Java programs more efficient, as the programmer does not need to worry about managing memory.