Assignment Questions 5

Q1. What is Exception in Java?

Exception is an unwanted or unexpected event, which occurs during the execution of a program, i.e. at run time, that disrupts the normal flow of the program's instructions. Exceptions can be caught and handled by the program. When an exception occurs within a method, it creates an object. This object is called the exception object. It contains information about the exception, such as the name and description of the exception and the state of the program when the exception occurred.

Q2. What is Exception Handling?

Exception Handling in Java is one of the effective means to handle the runtime errors so that the regular flow of the application can be preserved. Java Exception Handling is a mechanism to handle runtime errors such as ClassNotFoundException, IOException, SQLException, RemoteException, etc.

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

Checked Exception	Unchecked Exception
Checked exceptions occur at compile time.	Unchecked exceptions occur at runtime.
The compiler checks a checked exception.	The compiler does not check these types of exceptions.
These types of exceptions can be handled at the time of compilation.	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 the sub-class of the exception class.	They are runtime exceptions and hence are not a part of the Exception class.

Here, the JVM needs the exception to catch and handle.	Here, the JVM does not require the exception to catch and handle.
Examples of Checked exceptions: • File Not Found Exception • No Such Field Exception • Interrupted Exception • No Such Method Exception • Class Not Found Exception	Examples of Unchecked Exceptions: • No Such Element Exception • Undeclared Throwable Exception • Empty Stack Exception • Arithmetic Exception • Null Pointer Exception • Array Index Out of Bounds Exception • Security Exception

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

Sr. no.	Basis of Differences	throw	throws
1.	Definition	Java throw keyword is used throw an exception explicitly in the code, inside the function or the block of code.	Java throws keyword is used in the method signature to declare an exception which might be thrown by the function while the execution of the code.
2.	Type of exception Using throw keyword, we can only propagate unchecked exception i.e., the checked exception cannot be propagated using throw only.	Using throws keyword, we can declare both checked and unchecked exceptions. However, the throws keyword can be used to propagate checked exceptions only.	

3.	Syntax	The throw keyword is followed by an instance of Exception to be thrown.	The throws keyword is followed by class names of Exceptions to be thrown.
4.	Declaration	throw is used within the method.	throws is used with the method signature.
5.	Internal implementation	We are allowed to throw only one exception at a time i.e. we cannot throw multiple exceptions.	We can declare multiple exceptions using throws keyword that can be thrown by the method. For example, main() throws IOException, SQLException.

Q5. What is multithreading in Java? mention its advantages

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.

Threads can be created by using two mechanisms :

1. Extending the Thread class
2. Implementing the Runnable Interface

Some of the benefits of multithreaded programming are given as follows –

- **Resource Sharing -** All the threads of a process share its resources such as memory, data, files etc. A single application can have different threads within the same address space using resource sharing.
- **Responsiveness** Program responsiveness allows a program to run even if part of it is blocked using multithreading. This can also be done if the process is performing a lengthy operation. For example - A web browser with multithreading can use one thread for user contact and another for image loading at the same time.
- **Utilization of Multiprocessor Architecture -** In a multiprocessor architecture, each thread can run on a different processor in parallel using multithreading. This increases concurrency of the system. This is in direct contrast to a single processor system, where only one process or thread can run on a processor at a time.

- **Economy -** It is more economical to use threads as they share the process resources. Comparatively, it is more expensive and time-consuming to create processes as they require more memory and resources. The overhead for process creation and management is much higher than thread creation and management.

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

```
class MyException extends Exception {
   public MyException(String s)
   {
      // Call constructor of parent Exception
      super(s);
   }
}

// A Class that uses above MyException
public class Main {
   // Driver Program
   public static void main(String args[])
   {
      try {
         // Throw an object of user defined exception
         throw new MyException("GeeksGeeks");
      }
      catch (MyException ex) {
         System.out.println("Caught");

         // Print the message from MyException object
         System.out.println(ex.getMessage());
      }
   }
}
```

Q7. How can you handle exceptions in Java?

Customized Exception Handling: Java exception handling is managed via five keywords: **try**, **catch**, **throw**, **throws**, and **finally**. Briefly, here is how they work. Program statements that you think can raise exceptions are contained within a try block. If an exception occurs within the try block, it is thrown. Your code can catch this exception (using catch block) and handle it in some rational manner. System-generated exceptions are automatically thrown by the Java run-time system. To manually throw an exception, use the keyword throw. Any exception that is thrown out of a method must be specified as such by a throws clause. Any code that absolutely must be executed after a try block completes is put in a finally block.

Q8. What is Thread in Java?

A thread in Java is the direction or path that is taken while a program is being executed. Generally, all the programs have at least one thread, known as the main thread, that is provided by the JVM or Java Virtual Machine at the starting of the program's execution. At this point, when the main thread is provided, the main() method is invoked by the main thread.

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

There are two ways to create a thread:

1. By extending Thread class
2. By implementing Runnable interface.

Q10.What do you mean by garbage collection?

Java garbage collection is the process by which Java programs perform automatic memory management. Java programs compile to bytecode that can be run on a Java Virtual Machine, or JVM for short. When Java programs run on the JVM, objects are created on the heap, which is a portion of memory dedicated to the program. Eventually, some objects will no longer be needed. The garbage collector finds these unused objects and deletes them to free up memory.