Week 3- Session 3

1. What is exception handling in Java and what are the advantages of exception handling?

Exception Handling is the technique of handling unexpected failures that could occur in a program so that the program does not terminate and normal execution flow is maintained

The following are some of the Advantages of using Exception Handling in Java:

1. The most important advantage of having an exception-handling technique is that it avoids abnormal program termination and the normal flow of the program is maintained.
2. Provides flexibility to the programmers to define/handle what should occur in cases of failures/errors thereby making the applications more robust and immune to unexpected scenarios.
3. Provides stable user experience in cases of failures by providing means to let the user know what made the program fail.

2. How are exceptions handled in Java?

In Java, exceptions could be handled in the following ways:

1. try-catch block: The try section holds the code that needs to be normally executed and it monitors for any possible exception that could occur. The catch block “catches” the exception thrown by the try block. It could consist of logic to handle failure scenario or the catch block could simply rethrow the exception by using the “throw” keyword.
2. finally block: Regardless of whether an exception has occurred or not, if we need to execute any logic, then we place it in the final block that is usually

associated with the try-catch block or just with the try block. The final block is not executed when System.exit (0) is present in either the try or catch block.

3. What is exception propagation in Java?

Exception propagation is a process where the compiler makes sure that the exception is handled if it is not handled where it occurs. If an exception is not caught where it occurred, then the exception goes down the call stack of the preceding method and if it is still not caught there, the exception propagates further down to the previous method. If the exception is not handled anywhere in between, the process continues until the exception reaches the bottom of the call stack. If the exception is still not handled in the last method, i.e, the main method, then the program gets terminated.

4. What is the difference between the throw and throws keywords in Java?

The throw keyword allows a programmer to throw an exception object to interrupt normal program flow. The exception object is handed over to the runtime to handle it. For example, if we want to signify the status of a task is outdated, we can create an OutdatedTaskException that extends the Exception class and we can throw this exception object as shown below:

```
If (task.getStatus ().equals ("outdated")) {
   throw new OutdatedTaskException("Task is outdated");
}
```

The throws keyword in Java is used along with the method signature to specify exceptions that the method could throw during execution. For example, a method could throw NullPointerException or FileNotFoundException and we can specify that in the method signature as shown below:

```
public void someMethod () throws NullPointerException,
FileNotFoundException {
   // do something
}
```

5. How are the keywords final, finally and finalize different from each other?

- **final keyword:** By using this keyword,

We can declare a variable as final (meaning, variable value cannot be changed).

We can declare a method as final (meaning, that method cannot be overridden).

We can declare a class as final (meaning, that class cannot be extended).

- **finally keyword:** This is used in conjunction with the try-catch block or the try block where we want to run some logic whether or not an exception has occurred.
- **finalize keyword**: This is a method called by the Garbage Collector just before destroying the objects no longer needed in the program.

6**. How many types of exception can occur in a Java program?**

There are mainly two types of exceptions: checked and unchecked. Here, an error is considered as the unchecked exception. According to Oracle, there are three types of exceptions:

- Checked Exception: Checked exceptions are the one which are checked at compile-time. For example, SQLException, ClassNotFoundException, etc.

- Unchecked Exception: Unchecked exceptions are the one which are handled at runtime because they cannot be checked at compile-time. For example, ArithmaticException, NullPointerException, ArrayIndexOutOfBoundsException, etc.

- Error: Error cause the program to exit since they are not recoverable. For Example, OutOfMemoryError, AssertionError, etc.

7. What is Exception Handling?

Exception Handling is a mechanism that is used to handle runtime errors. It is used primarily to handle checked exceptions. Exception handling maintains the normal flow of the program. There are mainly two types of exceptions: checked and unchecked. Here, the error is considered as the unchecked exception.

8. What is the difference between Checked Exception and Unchecked Exception?

1) **Checked Exception**

The classes that extend Throwable class except RuntimeException and Error are known as checked exceptions, e.g., IOException, SQLException, etc. Checked exceptions are checked at compile-time.

2**) Unchecked Exception**

The classes that extend RuntimeException are known as unchecked exceptions, e.g., Arithmetic Exception, NullPointerException, etc. Unchecked exceptions are not checked at compile-time.

9. What is the base class for Error and Exception?

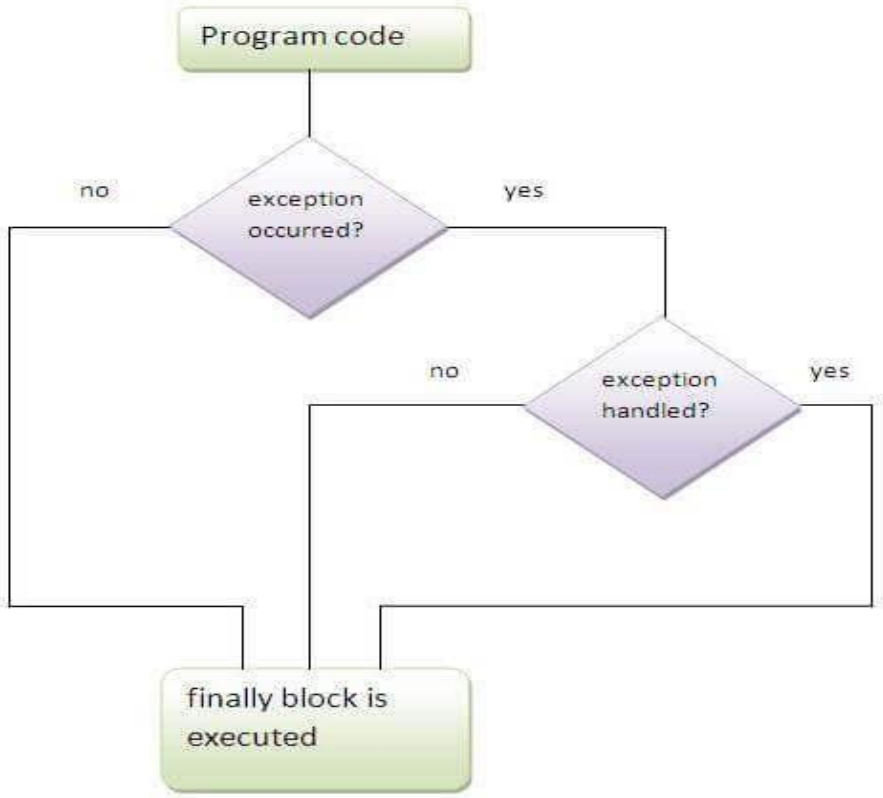
The Throwable class is the base class for Error and Exception.

10. Is it necessary that each try block must be followed by a catch block?

It is not necessary that each try block must be followed by a catch block. It should be followed by either a catch block OR a finally block. So whatever exceptions are likely to be thrown should be declared in the throws clause of the method.

11. What is finally block?

The "finally" block is used to execute the important code of the program. It is executed whether an exception is handled or not. In other words, we can say that finally block is the block which is always executed. Finally block follows try or catch block. If you don't handle the exception, before terminating the program, JVM runs finally block, (if any). The finally block is mainly used to place the clean-up code such as closing a file or closing a connection. Here, we must know that for each try block there can be zero or more catch blocks, but only one finally block. The finally block will not be executed if program exits (either by calling System.exit () or by causing a fatal error that causes the process to abort).



12. Can finally block be used without a catch?

Yes, According to the definition of finally block, it must be followed by a try or catch block, therefore, we can use try block instead of catch.

13. What is the difference between throw and throws?

throw keyword	throws keyword
1) The throw keyword is used to throw an exception explicitly.	The throws keyword is used to declare an exception.
2) The checked exceptions cannot be propagated with throw only.	The checked exception can be propagated with throws

3) The throw keyword is followed by an instance.	The throws keyword is followed by class.
4) The throw keyword is used within the method.	The throws keyword is used with the method signature.
5) You cannot throw multiple exceptions.	You can declare multiple exceptions, e.g., public void method () throws IOException, SQLException.

14. Why exception handling is required in Java?

Exception handling is the process of responding to unwanted or unexpected events when a computer program runs. Exception handling deals with these events to avoid the program or system crashing, and without this process, exceptions would disrupt the normal operation of a program.

15. What is the difference between final, finally and finalize in Java?

final	finally	finalize()
final is a keyword in Java which is used to make a variable or a method or a class as unchangeable.	finally is a block in Java which is used for exception handling along with try and catch blocks.	finalize () method is a protected method of java.lang.Object class which is used to perform some clean up operations on an object before it is removed from the memory.

The value of a variable which is declared as final can't be changed once it is initialized.	finally block is always executed whether an exception is occurred or not and occurred exception is handled or not.	This method is called by garbage collector thread before an object is removed from the memory.
A method declared as final can't be overridden or modified in the sub class and a class declared as final can't be extended.	Most of time, this block is used to close the resources like database connection, I/O resources etc soon after their use.	This method is inherited to every class you create in Java.

16. Does the finally block always get executed in the Java program?

The finally blocks are meant to be executed whenever there is an exception or not in the try-catch blocks. However, there is one scenario where this block does not get executed. It is when we use System.exit(0) in either try or catch block. This is because the System.exit(0) terminates the running JVM.