Assignment-14

1.what is exception?

Ans:

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.

2. what is exception handling?

Ans:

The Exception Handling in Java is one of the powerful mechanism to handle the runtime errors so that the normal flow of the application can be maintained.

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

3. Exception Hierarchy?

Ans:

The hierarchy of Exceptions in the Java programming language begins with the

Throwable class – which comes from the Object class and is its direct subclasswhileThe Exception class presents all This Throwable class further branches into two subclasses – Error and Exception

4. What is checked exceptions and how will you handle it?

Ans:

The classes that directly inherit the Throwable class except RuntimeException and Error are known as checked exceptions.For example, IOException, SQLException, Checked exceptions are checked at compile-time.
These types of exceptions need to be handled during the compile time of the program

Handled by:

These exceptions can be handled by the try-catch block or by using throws keyword otherwise the program will give a compilation error.

5. what is unchecked exceptions and how will you handle it

Ans:

These types of Exceptions occur during the runtime of the program.
These are the exceptions that are not checked at a compiled time by the compiler. In Java exceptions under Error and Runtime Exception classes are unchecked exceptions, This Exception occurs due to bad programming.

Errors class Exceptions like StackOverflow, OutOfMemoryError exception, etc are difficult to handle
Runtime Exceptions like IndexoutOfBoundException, Nullpointer Exception, etc can be handled with the help of try-Catch Block

6. types of handling checked exceptions?

Ans:

Checked exceptions are the subclass of the Exception class.
*These types of exceptions need to be handled during the compile time of the program.
*These exceptions can be handled by the try-catch block or by using throws keyword otherwise the program will give a compilation error.
ClassNotFoundException, IOException, SQLException etc are the examples of the checked exceptions.

7. Why do we need the finally block?

Ans:

- The finally block in java is used to put important codes such as clean up code e.g. closing the file or closing the connection.
- The finally block executes whether exception rise or not and whether exception handled or not.
- A finally contains all the crucial statements regardless of the exception occurs or not.
- But finally is useful for more than just exception handling — it allows the programmer to avoid having cleanup code accidentally bypassed by a return , continue , or break .
- Putting cleanup code in a finally block is always a good practice, even when no exceptions are anticipated.

- The finally keyword is used to execute code (used with exceptions - try.. catch statements) no matter if there is an exception or not.

- A finally block always executes, regardless of whether an exception is thrown. The following code example uses a try / catch block to catch an ArgumentOutOfRangeException

- The code opens a file and then executes statements that use the file; the finally -block makes sure the file always closes after it is used even if an exception was thrown

- It is not mandatory to include a finally block at all, but if you do, it will run regardless of whether an exception was thrown and handled by the try and catch parts of the block.

8. Is multiple catch blocks allowed?

Ans:

Yes, we can define one try block with multiple catch blocks in Java.

Every try should and must be associated with at least one catch block.
*Whenever an exception object is identified in a try block and if there are multiple catch blocks then the priority for the catch block would be given based on the order in which catch blocks are have been defined.
*Highest priority would be always given to first catch block.
*If the first catch block cannot handle the identified exception object then it considers the immediate next catch block

Before Java 7, we had to catch only one exception type in each catch block. So, whenever we needed to handle more than one specific exception but take some action for all exceptions, we had to have more than one catch block containing the same code.

In the following code, we have to handle two different exceptions but take the same action for both. So we needed to have two different catch blocks as of Java 6.0.

// A Java program to demonstrate that we needed
// multiple catch blocks for multiple exceptions
// prior to Java 7

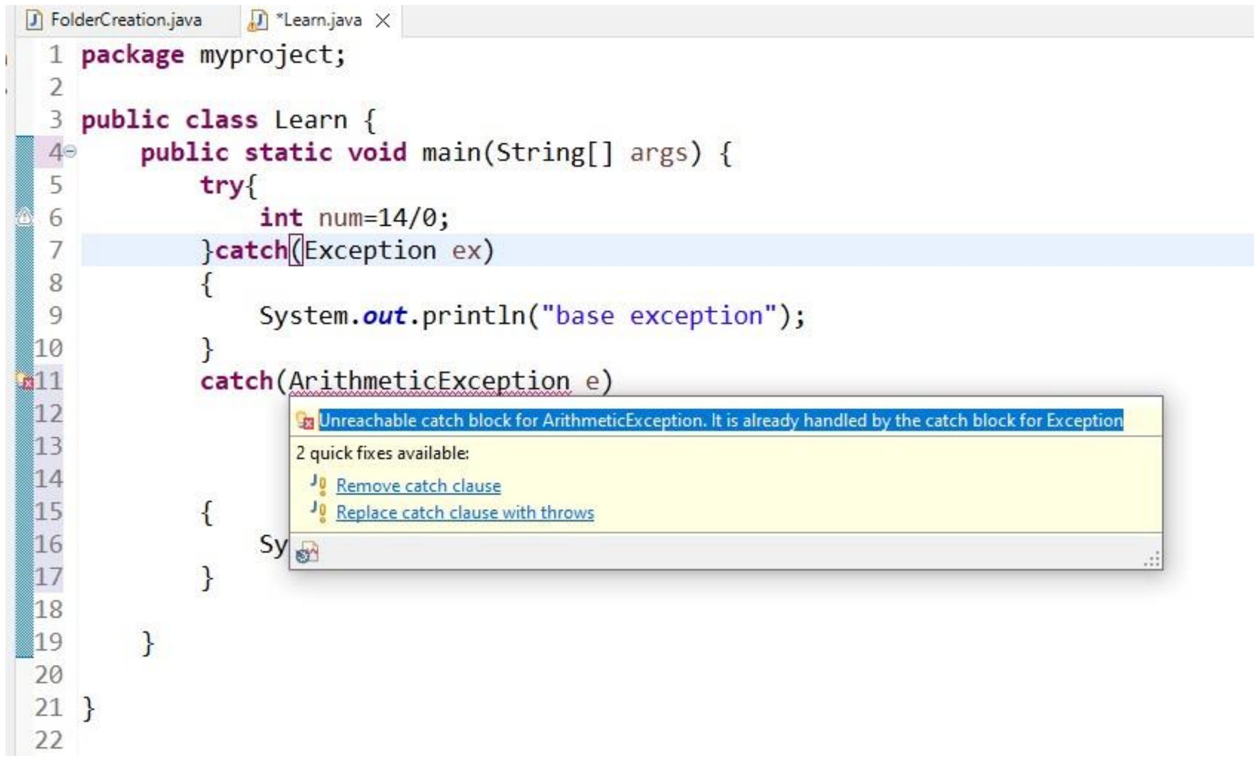
9. What is the output of below program try{

int num=14/0; }catch(Exception ex)

{

Sop("base exception");

}catch(ArthmeticException e){sop("child exception") Ans:


FolderCreation.java *Learn.java

```
package myproject;

public class Learn {
    public static void main(String[] args) {
        try{
            int num=14/0;
        }catch(Exception ex)
        {
            System.out.println("base exception");
        }
        catch(ArithmeticException e)



        {
            Sy
        }

    }

}

```

Unreachable catch block for ArithmeticException. It is already handled by the catch block for Exception

2 quick fixes available:

Remove catch clause

Replace catch clause with throws

10. what is try with resources?

Ans:

- The try-with-resources Statement
- The try-with-resources statement is a try statement that declares one or more resources.
- A resource is an object that must be closed after the program is finished with it.
- The try-with-resources statement ensures that each resource is closed at the end of the statement.
- Any object that implements java.lang.AutoCloseable, which includes all objects which implement java.io.Closeable, can be used as a resource.

11. difference between the closeable and autocloseable?

Ans:

Closeable	**AutoCloseable**
1. Closeable was introduced with JDK 5	**1.** AutoCloseable was introduced with JDK 7+.
2. Closeable extends IOException	**2.** AutoCloseable extends Exception.
3. Closeable interface is idempotent (calling close() method more than once does not have any side effects)	**3.** AutoCloseable does not provide this feature.
4. AutoCloseable was specially introduced to work with trywith-resources statements.	**4.** Since Closeable implements AutoCloseable, therefore any class that implements Closeable also implements AutoCloseable interface and can use the trywith resources to close the files.

12. which method is present in closeable?

Ans:

Modifier	**Type Method**
void	close()

Description

Detail

close

void close()

throws IOException

Closes this stream and releases any system resources associated with it. If the stream is already closed then invoking this method has no effect.

Specified by: close in interface

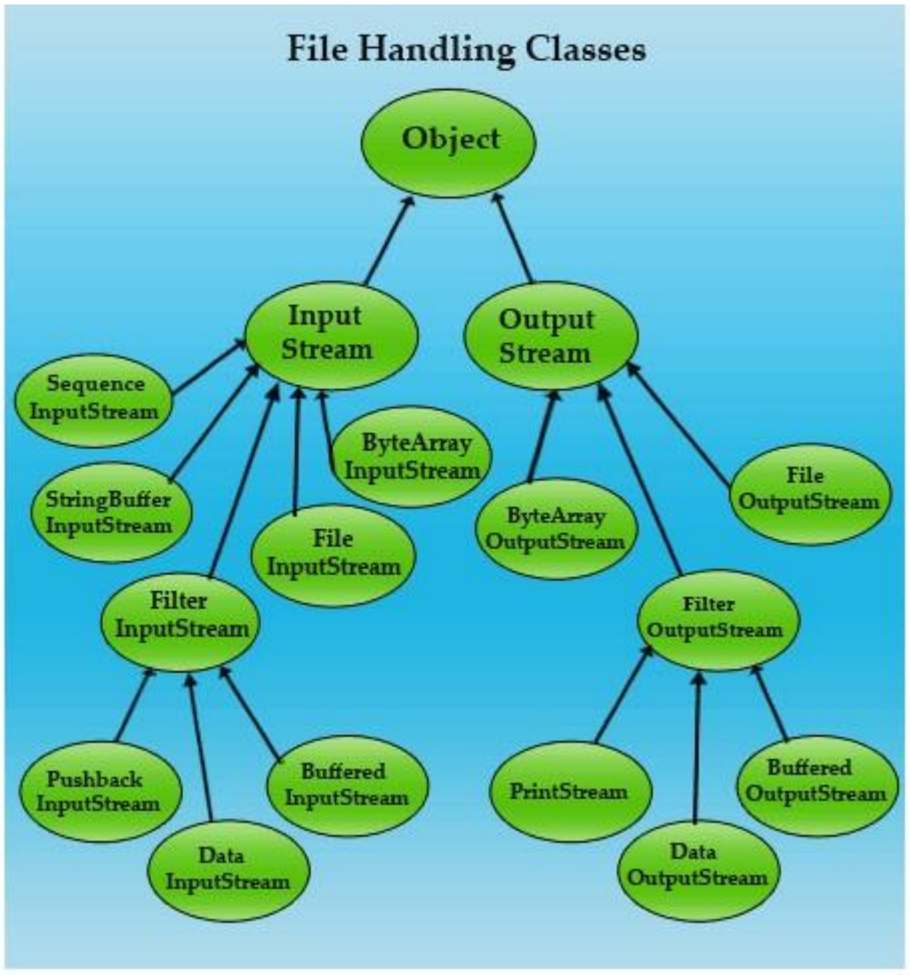
13. name few classes implementing the closeable?

Ans:

- AbstractSelectableChannel
- AbstractSelector
- BufferedReader
- BufferedWriter
- BufferedInputStream
- BufferedOutputStream
- CheckedInputStream
- CheckedOutputStream

14. Types of filehandling classes?

Ans:



15. What is the return type of read method() in FileReader class?

Ans:

The read() method of FileReader class in Java is used to read and return a single character in the form of an integer value that contains the character's char value. The character read as an integer in the range of 0 to 65535 is returned by this function. If it returns -1 as an int number, it means that all of the data has been read and that FileReader may be closed.

Syntax:
public abstract int read()

Returns: The read() method returns a single character in the form of an integer value that contains the character's char value. It returns -1 when all of the data has been read and that FileReader may be closed.

16. Name the exceptions come across while working with filehandling?

Ans:

Hierarchy of Java Exception classes

- Checked Exception.
- Unchecked Exception.

Error When we run this program, if the file test. txt does not exist, FileInputStream throws a FileNotFoundException which extends the IOException class.
The findFile() method specifies that an IOException can be thrown.
The main() method calls this method and handles the exception if it is thrown.

17. FileReader constructor throws which exception?

Ans:

Creates a new FileReader, given the name of the file to read from.
Parameters:
fileName - the name of the file to read from **Throws:**
FileNotFoundException - if the named file does not exist, is a directory rather than a regular file, or for some other reason cannot be opened for reading. FileReader
public FileReader(File file)
throws FileNotFoundException
Creates a new FileReader, given the File to read from.
Parameters:
file - the File to read from **Throws:**
FileNotFoundException - if the file does not exist, is a directory rather than a regular file, or for some other reason cannot be opened for reading

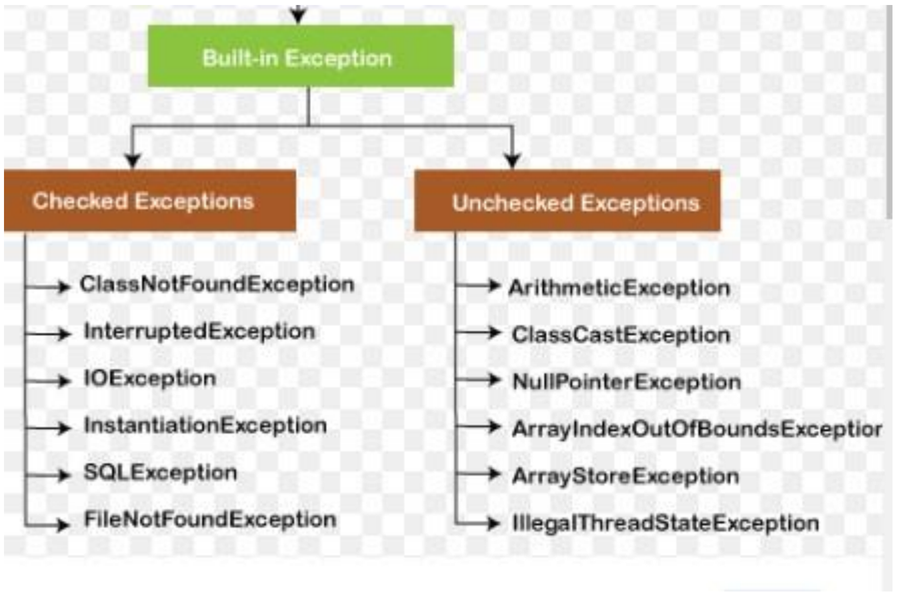
18. read() method and close() method throws which exception?

Ans:

close() and read() methods? Explanation: Both close() and read() method throw **IOException**

19. Name few unchecked exceptions and checked exceptions?

Ans:


Built-in Exception
Checked Exceptions
ClassNotFoundException
InterruptedException
IOException
InstantiationException
SQLException
FileNotFoundException
Unchecked Exceptions
ArithmeticException
ClassCastException
NullPointerException
ArrayIndexOutOfBoundsExceptior
ArrayStoreException
IllegalThreadStateException