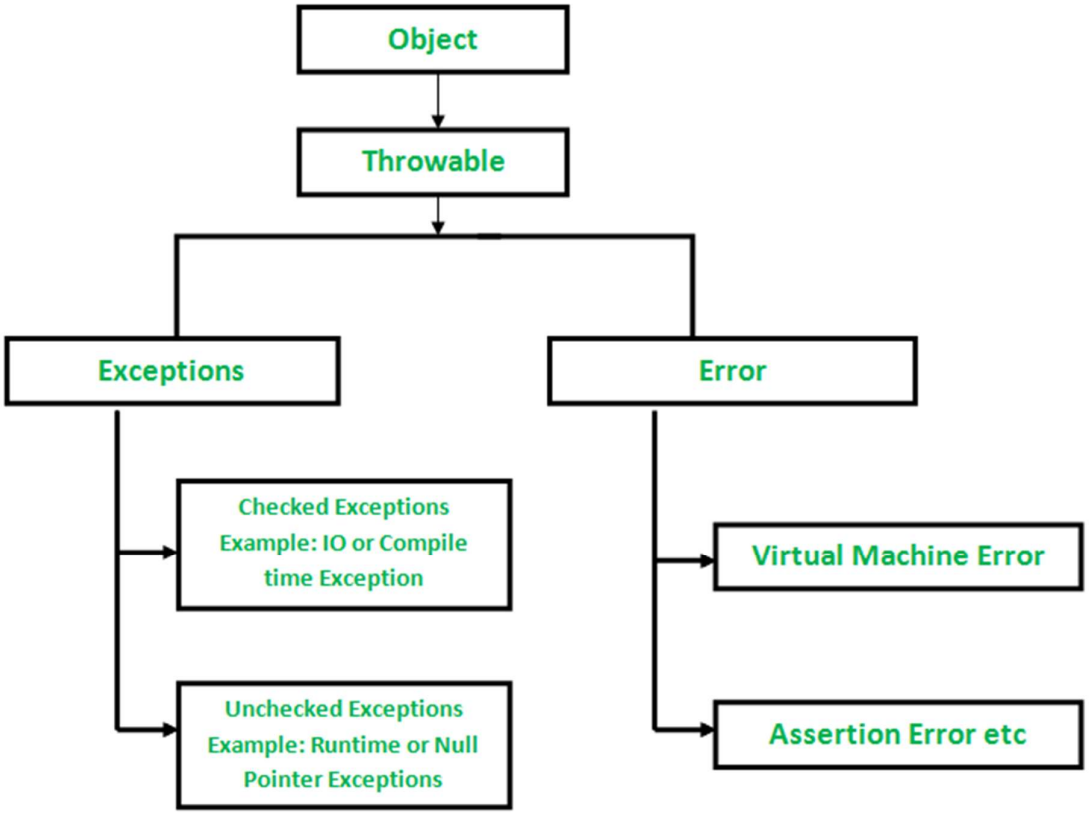
1.what is exception?
Ans:
An event that occurs during the execution of a program that disrupts the normal flow of instructions is called an exception.
Example: public static void Main()

2.what is exception handling?
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.

3.Exception Hierarchy?



4.What is checked exceptions and how will you handle it?
A checked exception is caught at compile time whereas a runtime or unchecked exception is, as it states, at runtime. A checked exception must be handled either by re-throwing or with a try catch block, whereas an unchecked isn't required to be handled.

5.what is unchecked exceptions and how will you handle it?
Ans:
An unchecked exception (also known as an runtime exception) in Java is something that has gone wrong with the program and is unrecoverable. Just because this is not a compile time

exception, meaning you do not need to handle it, that does not mean you don't need to be concerned about it.

6.types of handling checked exceptions?
A checked exception must be handled either by re-throwing or with a try catch block, whereas an unchecked isn't required to be handled. A runtime exception is a programming error and is fatal whereas a checked exception is an exception condition within your code's logic and can be recovered or re-tried from.

7.Why do we need the finally block?

A Finally block is useful for running any code that must execute even if there is an exception. Control is passed to the Finally block regardless of how the Try...Catch block exits. The code in a Finally block runs even if your code encounters a Return statement in a Try or Catch block.

8.Is multiple catch blocks allowed?
Ans: yes it is allowed.

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")}
```

Output:
Base exception

10.what is try with resources?

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.

11.difference between the closeable and autocloseable?
Ans:
Both are used to release resources after the try-with-resources block ends. The Closeable is the old one that exists only to preserve backward compatibility. The method close() in Closeable interfaces throws IOException and AutoCloseable.close() method throws Exception class.

12.which method is present in closeable?

Ans: A Closeable is a source or destination of data that can be closed. The close method is invoked to release resources that the object is holding (such as open files).

13.name few classes implementing the closeable?
Ans:

AbstractSelectableChannel
AbstractSelector
BufferedReader
BufferedWriter
BufferedInputStream
BufferedOutputStream
CheckedInputStream
CheckedOutputStream

14.Types of filehandling classes?
Bytestream class
Charstream class

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

17. FileReader constructor throws which exception?
Ans:
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.

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

19.Name few unchecked exceptions and checked exceptions?
Ans: Common checked exceptions include IOException, DataAccessException, InterruptedException, etc. Common unchecked exceptions include ArithmeticException, InvalidClassException, NullPointerException, etc.