UNIT III – EXCEPTION HANDLING AND I/O

Part A – Question Bank

1. What is exception?

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

2. What is error?

An Error indicates that a non-recoverable condition has occurred that should not be caught. Error, a subclass of Throwable, is intended for drastic problems, such as OutOf-MemoryError, which would be reported by the JVM itself.

3. Which is super class of Exception?

"Throwable", the parent class of all exception related classes.

4. What are the advantages of using exception handling?

Exception handling provides the following advantages over "traditional" error management techniques:

Separating Error Handling Code from "Regular" Code. Propagating Errors Up the Call Stack.

Grouping Error Types and Error Differentiation.

5. What are the types of Exceptions in Java

There are two types of exceptions in Java, unchecked exceptions and checked exceptions.

- Checked exceptions: A checked exception is some subclass of Exception (or Exception itself), excluding class RuntimeException and its subclasses. Each method must either handle all checked exceptions by supplying a catch clause or list each unhandled checked exception as a thrown exception.
- Unchecked exceptions: All Exceptions that extend the RuntimeException class are unchecked exceptions. Class Error and its subclasses also are unchecked.

6. Why Errors are not checked?

An unchecked exception classes which are the error classes (Error and its subclasses) are exempted from compile-time checking because they can occur at many points in the program and recovery from them is difficult or impossible. A program declaring such exceptions would be pointlessly.

7. How does a try statement determine which catch clause should be used to handle an exception?

When an exception is thrown within the body of a try statement, the catch clauses of the try statement are examined in the order in which they appear. The first catch clause that is capable of handling the exception is executed. The remaining catch clauses are ignored.

8. What is the purpose of the finally clause of a try-catch-finally statement?

The finally clause is used to provide the capability to execute code no matter whether or not an exception is thrown or caught.

9. What is the difference between checked and unchecked Exceptions in Java?

All predefined exceptions in Java are either a checked exception or an unchecked exception. Checked exceptions must be caught using try catch () block or we should throw the exception using throws clause. If you don't, compilation of program will fail.

10. What is the difference between exception and error?

The exception class defines mild error conditions that your program encounters. Exceptions can occur when trying to open the file, which does not exist, the network connection is disrupted, operands being manipulated are out of prescribed ranges, the class file you are interested in loading is missing. The error class defines serious error conditions that you should not attempt to recover from. In most cases it is advisable to let the program terminate when such an error is encountered.

11. What is the catch or declare rule for method declarations?

If a checked exception may be thrown within the body of a method, the method must either catch the exception or declare it in its throws clause.

12. When is the finally clause of a try-catch-finally statement executed?

The finally clause of the try-catch-finally statement is always executed unless the thread of execution terminates or an exception occurs within the execution of the finally clause.

13. What if there is a break or return statement in try block followed by finally block?

If there is a return statement in the try block, the finally block executes right after the return statement encountered, and before the return executes.

14. Write short notes on throwable class.

The Throwable class is the superclass of all errors and exceptions in the Java language. It contains a snapshot of the execution stack of its thread at the time it was created. It can also contain a message string that gives more information about the error. The Exception class inherits all the methods from the class Throwable.

15. Write short notes on Exception class.

Java allows the user to create their own exception class which is derived from built-in class Exception. The Exception class inherits all the methods from the class Throwable.

Two commonly used constructors of Exception class are:

- Exception() - Constructs a new exception with null as its detail message.
- Exception(String message) - Constructs a new exception with the specified detail message.

Syntax:

```
class User_defined_name extends Exception{
……..
}
```

16. How is custom exception created?

The user defined exception is created by extending the Exception class or one of its sub-classes.

Example:

```
class MyException extends Exception {
public MyException(){super();}
Public MyException(String s){super(s);}
}
```

17. What are the different ways to handle exceptions?

There are two ways to handle Exceptions

- Wrapping the desire code in a try block followed by a catch block to catch the exception
- List the desired exception in the throws clause of the method and let the caller of the method handle those exceptions.

18. What do you mean by chained exceptions?

Chained Exceptions allows to relate one exception with another exception, i.e one exception describes cause of another exception.

Throwable methods that supports chained exceptions are:

1. getCause() method :- This method returns actual cause of an exception.
2. initCause(Throwable cause) method :- This method sets the cause for the calling exception.

19. Write short notes on stackTraceElement.

The StackTraceElement class element represents a single stack frame which is a stack trace when an exception occurs. Extracting stack trace from an exception could provide useful information such as class name, method name, file name, and the source-code line number. This creates a stack trace element representing the specified execution point.

getStackTrace( ) - returns an array of StackTraceElements

20. List any three common run time errors.

Exception	Description
ArithmeticException	Arithmetic error, such as divide-by-zero
ArrayIndexOutOfBoundsException	Array index is out-of-bounds
IllegalThreadStateException	Requested operation not compatible with current thread state

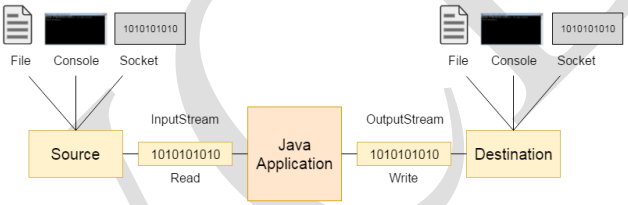
21. Differentiate throw and throws in Java.

throw	throws
Java throw keyword is used to explicitly throw an exception.	Java throws keyword is used to declare an exception.
Checked exception cannot be propagated using throw only.	Checked exception can be propagated with throws.
Throw is followed by an instance.	Throws is followed by class.
Throw is used within the method.	Throws is used with the method signature.
You cannot throw multiple exceptions.	You can declare multiple exceptions e.g. public void method()throws IO Exception, SQL Exception.

void m(){
throw new IOException("Error");
}
	void m() throws IOException{
	//code
	}

22. Define Stream.

A stream can be defined as a sequence of data. The InputStream is used to read data from a source and the OutputStream is used for writing data to a destination.



23. List the type of streams supported in Java.

1. ***Byte Stream :*** It is used for handling input and output of 8 bit bytes. The frequently used classes are FileInputStream and FileOutputStream.
2. ***Character Stream :*** It is used for handling input and output of characters. Character stream uses 16 bit Unicode. The frequently used classes are FileReader and File Writer.

24. List out the predefined streams available in Java.

Java provides the following three standard streams −

- Standard Input − refers to the standard InputStream which is the keyboard by default. This is used to feed the data to user's program and represented as **system.in**.
- Standard Output − refers to the standard OutputStream by default,this is console and represented as **system.out**.
- Standard Error − This is used to output the error data produced by the user's program and usually a computer screen is used for standard error stream and represented as **system.err**.

25. What is Byte stream in Java? List some of the Bytestream classes.

The byte stream classes provide a rich environment for handling byte-oriented I/O.

List of Byte Stream classes

- Byte Array Input Stream
- Byte Array Output Stream
- Filtered Byte Streams
- Buffered Byte Streams

26. What is character stream in Java? List some of the characterstream classes.

The Character Stream classes provide a rich environment for handling character-oriented I/O.

List of Character Stream classes

- File Reader
- File Writer
- Char Array Reader
- Char Array Writer

27. What are the steps involved in stream i/o operations?

1. Open an input/output stream associated with a physical device (e.g., file, network, console/keyboard), by constructing an appropriate I/O stream instance.
2. Read from the opened input stream until "end-of-stream" encountered, or *write* to the opened output stream (and optionally flush the buffered output).
3. Close the input/output stream.

28. Write about read() method.

- returns the input byte read as an int in the range of 0 to 255, or
- returns -1 if "end of stream" condition is detected, or
- throws an IOException if it encounters an I/O error.
- The read() method returns an int instead of a byte, because it uses -1 to indicate end-of-stream.
- The read() method blocks until a byte is available, an I/O error occurs, or the "end-of-stream" is detected.

29. What are the two variations of read() method?

public int read(byte[] bytes, int offset, int length) throws IOException

public int read(byte[] bytes) throws IOException

30. What are the two variations of write() method?

public void write(byte[] bytes, int offset, int length) throws IOException

public void write(byte[] bytes) throws IOException

31. What's the difference between println(), print() and printf()?

print() - prints string inside the quotes
println() - prints string inside the quotes similar like print() method. Then the cursor moves to the beginning of the next line.
printf() - it provides string formatting (similar to printf in C/C++ programming).

32. Write about Fileinputstream.

- This stream is used for reading data from the files.
- Objects can be created using the keyword new and there are several types of constructors available.
- The two constructors which can be used to create a FileInputStream object:

i) Following constructor takes a file name as a string to create an input stream object to read the file:

```
OutputStream f = new FileOutputStream("filename ");
```

ii) Following constructor takes a file object to create an input stream object to read the file. First we create a file object using File() method as follows:

```
File f = new File("filename ");
InputStream f = new FileInputStream(f);
```

33. Write about Fileoutputstream.

- FileOutputStream is used to create a file and write data into it.
- The stream would create a file, if it doesn't already exist, before opening it for output.
- The two constructors which can be used to create a FileOutputStream object:

i) Following constructor takes a file name as a string to create an input stream object to write the file:

```
OutputStream f = new FileOutputStream("filename ");
```

ii) Following constructor takes a file object to create an output stream object to write the file. First, we create a file object using File() method as follows:

```
File f = new File("filename ");
OutputStream f = new FileOutputStream(f);
```

PART -B

1. Explain exception handling in Java with an example. (8)(CS1261 MAY /JUNE 2016)
2. Explain with an example the exception handling feature in Java.(8)

(CS1261 NOV /DEC 2016) (CS2311 APR/MAY-2015)

3. Explain I/O streams with suitable examples. (8) (IT2301 APR/MAY-2015)
4. Discuss about the Java error handling mechanism? What is the difference between "un-checked exceptions" and "checked exceptions"? What is the implication of catching all the exceptions with the type "Exception"? (8) (IT2301 APR/MAY-2015)
5. How are exceptions handled in Java? Elaborate with suitable examples. (8)

(IT2301 MAY/JUNE-2014)

6. Describe about different input and output streams and their classes. (16)

(IT2301 NOV/DEC-2012)

7. Discuss about throwing and catching exceptions. (16) (IT2301 NOV/DEC-2012)
8. Explain the concept of streams and its byte stream classes in detail.

(CS2311 MAY/JUNE-2014)

9. Explain the use of File stream classes and file modes. Write an example program for file manipulation. (16) (CS1361 NOV/DEC-2014)
10. Describe different types of exceptions. Write a program for handling Array out of bounds Exception. (16) (CS1361 NOV/DEC-2014)
11. Write a Java program that asks the user for a number and displays ten times the number. Also the program must throw an exception when the user enters a value greater than 10. (16) (CS1361 MAY/JUNE-2013)