UNIT - 3

**EXCEPTION HANDLING AND I/O**

Question Bank

**PART – A**

**2 – Marks**

1. **Define Exceptions.**

In Java, an exception is an event that disrupts the normal flow of the program. It throws an exception object at runtime and it is cached and processed by handling mechanisms. We need to handle the exceptions to make the system in safe sate.

1. **List the types of exception.**

There are three types of exceptions

* Checked Exceptions [Built in Exception, User defined Exceptions]
* Unchecked Exceptions. [Found at run time]
* Errors. [Found at run time but we can’t handle].

1. **List some causes for exceptions.**

* A user has entered an invalid data.
* A file that needs to be opened cannot be found.
* A network connection has been lost in the middle of communications or the JVM has run out of memory.
* Programming Errors.
* Physical Resource Failure [Hard Drive Failure].

1. **Distinguish between Exception and Error.**

|  |  |
| --- | --- |
| **Exceptions** | **Errors** |
| It is encountered at either compile time or run time. | Mostly encountered at runtime |
| We can handle the situation and proceed with execution if proper handling mechanism is deployed. | We can’t handle the situation programs may terminate with executing safe termination procedures. |

1. **What is use of throws keyword in exception handling?**

Throws is a keyword that indicates the following exceptions may rise but program won’t handle. It helps to avoid creating exception handling blocks like [try and catch] example IOException. But is not recommended by the developers.

1. **What are all the blocks used in exception handling?**

***try block:***

It is a block contains critical code [The code may rise and exception]

***catch block:***

It receives any type of inbuilt or user defined exception object and handling codes are available to handle the exceptions.

1. **List some any three build-in exceptions with its descriptions.**
   * ***ArithmeticException***
     + Arithmetic error, such as divide-by-zero.
   * ***ArrayIndexOutOfBoundsException***
     + Array index is out-of-bounds.
   * ***NegativeArraySizeException***
     + Array created with a negative size.
2. **Give short note on StackTraceElement class.**

StackTraceElement is a class that helps to store one stack frame. Stack frame consist of methods calls and values returned. One program execution may generate more than one StackTraceElement objects. getStackTrace() is method that returns the StackTraceElement object at any point of run time. It helps to debug the program for developers.

1. **What is use of finally block?**

The finally is block is an optional block when we using exception handling mechanism. Finally block contains fail safe codes for safe program termination. It contains close code for external recourses like streams and files. Exception may or may not occur but in both the situations the finally block gets executed without fail.

1. **Write an example program to handle ArrayIndexOutOfBoundsException.**

import java.io.\*;

class Main {

public static void main(String[] args){

int[] elements = new int[5];

try {

for(int i=0;i<10;i++){

elements[i] = i;

}

}

catch(ArrayIndexOutOfBoundsException e){

System.out.println("Array reached its maximum size");

System.out.println("Last element inserted in the array is :

"+elements[4]);

}

}

}

***Output:***

Array reached its maximum size

Last element inserted in the array is: 4

1. **Define Streams.**

Streams are represented as sequence of data. Here streams are the class that helps to perform read and write operations data from source to destination. There are several types of streams are available. Standard Streams, Character Streams, Byte Streams are the important streams in java.

1. **List the standard stream available in java.**

There are three types of standard streams available.

* Sytem.in – Helps to get the inputs from user.
* Sytem.out – Help to produce outputs.
* System.err – Helps to provide error information’s.

1. **Distinguish between Character and Byte Streams.**

|  |  |
| --- | --- |
| **Character Streams** | **Byte Streams** |
| Character streams are automatically reads the data as character by character [16 - bits]. | Byte streams are reading data as byte by byte [8 - bits] |
| It suitable to perform read and write operations in text files. Eg: input.txt | It suitable to perform read and write operations in binary files. Eg: input.jpg |
| Eg: FileReader, FileWriter | Eg: FileInputStream, FileOutputStream |

1. **Give a short note of Console class.**

* The Java.io.Console class provides methods to access the character-based console device, if any, associated with the current Java virtual machine.
* it is used to read from and write to the console, if one exists.
* Console is primarily a convenience class because most of its functionality is available through System.in and System.out.
* However, its use can simplify some types of console interactions, especially when reading strings from the console.

1. **Give a short note of Scanner class.**

* Java Scanner class comes under the java.util package. Java has various ways to read input from the keyboard, the java.util.Scanner class is one of them.
* The Java Scanner class breaks the input into tokens using a delimiter that is whitespace by default. It provides many methods to read and parse various primitive values.
* Java Scanner class is widely used to parse text for string and primitive types using a regular expression.

**PART – B**

**13 - Marks**

1. Write a detailed note of exception handling in java with an example.
2. Explain about user defined exception with an example.
3. Write a program to maintain student records with proper exception handling mechanisms.
4. Explain about streams in IO operations.
5. Write a program to copy binary data from one file to another file with proper exception handling.
6. Write a program to perform word count in text file.
7. Write a program to read contents from one file and convert all the words into uppercase and write it onto another file.