CSE 215L: Programming Language II Lab
Faculty: Silvia Ahmed, Sec – 4, 5
Lab instructor: Marufa Ferdausi
Lab 12 – Fall 2019

Objective:
After today's lab, the students should be able:

- To get an overview of exceptions and exception handling
- To write a **try-catch** block to handle exceptions
- To use the **finally** clause in a **try-catch** block
- To discover file/directory properties, to delete and rename files/directories, and to create directories using the **File** class

Exception

```
try {
Code to run;
A statement or a method that may
throw an exception;
More code to run;
}
catch (type ex) {
Code to process the exception;
}
```

```
public void myMethod()
throws Exception1, Exception2, ...,
ExceptionN
```

```
try {
statements;
}
catch (TheException ex) {
handling ex;
finally {
finalStatements;
}
}
```

Task – 1
(***InputMismatchException***) Write a program that prompts the user to read two integers and displays their sum. Your program should prompt the user to read the number again if the input is incorrect.

Task – 2
(***ArrayIndexOutOfBoundsException***) Write a program that meets the following requirements:
■ Creates an array with **100** randomly chosen integers.
■ Prompts the user to enter the index of the array, then displays the corresponding element value. If the specified index is out of bounds, display the message **Out of Bounds**.

Task – 3
(***NumberFormatException***) Write the **bin2Dec(String binaryString)** method to convert a binary string into a decimal number. Implement the **bin2Dec** method to throw a **NumberFormatException** if the string is not a binary string.

Task – 4

(*Count characters, words, and lines in a file*) Write a program that will count the number of characters, words, and lines in a file. Words are separated by whitespace characters. The file name should be passed as a command-line argument, as shown in Figure.

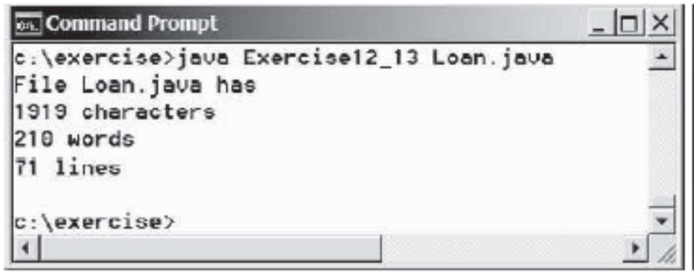

FIGURE The program displays the number of characters, words, and lines in the given file.

Task – 5

(*Process scores in a text file*) Suppose that a text file contains an unspecified number of scores separated by blanks. Write a program that prompts the user to enter the file, reads the scores from the file, and displays their total and average.

Task – 6

(*Write/read data*) Write a program to create a file named **Exercise12_15.txt** if it does not exist. Write **100** integers created randomly into the file using text I/O. Integers are separated by spaces in the file. Read the data back from the file and display the data in increasing order.