**JAVA Programming, CMSC 203**

**Worksheet 2**

**Module 11**

**Objectives**

* File I/O
* Exceptions

**Concept Questions**

1) A(n) \_\_\_\_\_\_\_\_ is an object that is generated in memory as the result of an error or an unexpected event.

A) exception handler

**B) exception**

C) default exception handler

D) error message

Answer: B

2) All of the exceptions that you will handle are instances of classes that extend this class.

A) RunTimeException

B) IOException

C) Error

**D) Exception**

Answer: D

3) All exceptions are instances of classes that extend this class.

A) RunTimeException

**B) Throwable**

C) Error

D) Exception

Answer: B

4) In a try/catch construct, after the catch statement is executed:

A) the program returns to the statement following the statement in which the exception occurred

B) the program terminates

C) **the program resumes at the statement that immediately follows the try/catch construct**

D) the program resumes at the first statement of the try statement

Answer: C

5) An exception's default error message can be retrieved using this method.

A) **getMessage()**

B) getErrorMessage()

C) getDefaultMessage()

D) getDefaultErrorMessage()

Answer: A

6) Why does the following code cause a compiler error?

try

{

number = Integer.parseInt(str);

}

catch (IllegalArgumentException e)

{

System.out.println("Bad number format.");

}

catch (NumberFormatException e)

{

System.out.println(str + " is not a number.");

}

A) Because you can have only one catch clause in a try statement

B) **Because NumberFormatException inherits from IllegalArgumentException. The code should handle NumberFormatException before IllegalArgumentException**

C) Because the Integer.parseInt method does not throw a NumberFormatException

D) Because the Integer.parseInt method does not throw an IllegalArgumentException

Answer: B

Diagram

Description automatically generated

7) If, within one try statement you want to have catch clauses of the following types, in which order should they appear in your program:

(1) Exception

(2) IllegalArgumentException

(3) RuntimeException

(4) Throwable

A) 1, 2, 3, 4

B) **2, 3, 1, 4**

C) 4, 1, 3, 2

D) 3, 1, 2, 4

Answer: B

8) When an exception is thrown:

A) it must always be handled by the method that throws it

B) the program terminates even if the exception is handled

C) **it must be handled by the program or by the default exception handler**

D) it may be ignored

Answer: C

9) The numeric classes' "parse" methods all throw an exception of this type if the string being converted does not contain a convertible numeric value.

A) **NumberFormatException**

B) ParseIntError

C) ExceptionMessage

D) FileNotFoundException

Answer: A

10) To read a file you will need a \_\_\_\_\_\_\_\_\_\_\_.

Answer: **Scanner object**

11) To write to a file you will need a \_\_\_\_\_\_\_\_\_\_\_\_\_\_.

Answer: **PrintWriter object**

*// if the file is already in existence or need to create use File object to create or read data from*

12) Write code which will write a string “Hello world” to a file called “Hello.txt”

ANSW:

**try** {

PrintWriter pw = **new** PrintWriter("Hello.txt");

pw.println("Hello World");

pw.close();

} **catch** (FileNotFoundException e) {

e.printStackTrace();

}

13) Write code which creates a File object which is linked to a file called “Hello.txt”

Answ:

File f = **new** File("Hello.txt");

18) Write a snippet of code which will read the data from file line by line and then output it in the console window

ANSW:

File f = **new** File("Hello.txt");

**try** {

Scanner sc = **new** Scanner(f);

**while**(sc.hasNextLine()){

String line = sc.nextLine();

System.***out***.println(line);

}

sc.close();

} **catch** (FileNotFoundException e) {

e.printStackTrace();

}

**Programming Questions**

* White the following program:
  + Prompt the user to enter a series of 10 numbers and save those numbers to a file called “Numbers.txt”. After that close the file.
  + Next, open the file “Numbers.txt”. While reading it line by line, add the number from the next line to the current line number.
    - EX: My file has numbers like this:

1

2

3

I will have two additions: 1+2 and 2+3

* + Save the results of the addition to a new file called “NumbersAdded.txt”
* import java.util.ArrayList;
* import java.util.Scanner;
* import java.io.\*;
* public class AddNumbers {
* public static void main(String[] args) throws IOException {
* // File name
* String fileName = "userNumbers.txt";
* // Get numbers from users
* getUserNumbers(fileName);
* // Add numbers
* addNumbersMethod(fileName);
* }
* /\*\*
* \* Get numbers from users
* \*
* \* @throws IOException
* \*/
* public static void getUserNumbers(String fileName) throws IOException {
* // create file to print to
* PrintWriter outFile = new PrintWriter(fileName);
* // read user input
* Scanner scan = new Scanner(System.in);
* int input = 0;
* do {
* System.out.print("Enter a string: ");
* input = scan.nextInt();
* // save numbers to file
* if (input != -1)
* outFile.println(input);
* } while (input != -1);
* outFile.close();
* System.out.println("Output file has been created: " + fileName);
* }
* /\*\*
* \* Add numbers in new file
* \*
* \* @param fileName
* \* @throws IOException
* \*/
* public static void addNumbersMethod(String fileName) throws IOException {
* // get the file to read from
* File file = new File(fileName);
* // read input from file
* Scanner input = new Scanner(file);
* // list to hold user numbers
* ArrayList<String> lines = new ArrayList<String>();
* PrintWriter newFile = new PrintWriter("userNumbersAdded.txt");
* while (input.hasNextLine()) {
* String value = input.nextLine();
* lines.add(value);
* }
* System.out.println(lines.toString());
* for (int i = 0; i < lines.size() - 1; i++) {
* int number = Integer.parseInt(lines.get(i)) + Integer.parseInt(lines.get(i + 1));
* newFile.println(number);
* }
* newFile.close();
* }
* }