Programming Fundamentals II Sec. 600

## Assignment #4

## Due date: 7/4/22 at 11:59 pm

1. (40 points) Briefly define (one to two sentences) each of the following ten terms.

1. Exception

*Runtime errors, occur when illegal operations performed. Most use no args constructor and message constructor*

1. try block

*used to catch and handle an exception that is thrown. Contains the code that is executed in normal circumstances.*

1. catch block

*can catch all subclasses of a specified exception. Contains the code that is executed in exceptional circumstances*

1. Checked exceptions

*Require the use of the catch block and throws clause. Must be included in code. All other exceptions except for RuntimeExceptions*

1. Unchecked exceptions

*Not required to be used in code. Must be corrected in code or critical system errors and consist of RuntimeExceptions, Error and their subclasses*

1. Call stack

*Holds the activation record and also known as an execution stack, runtime stack or machine stack*

1. finally clause

*used to execute code regardless of whether or not an exception is thrown*

1. File class

*Need files for permanent storage. Used to access or modify the properties of file/directory. Very particular and does not contain methods for reading and writing file contents*

1. PrintWriter class

*Used for writing text to a file and to create a file*

1. try-with-resources block

*Allows to automatically close resources in program*

2. (15 points) Briefly explain why Exceptions must be organized from most specific to most general.

*Exceptions organized from most specific to generic will handle any errors that are specific first then move on to the more general exceptions. When working with multiple catch blocks it is important to have your general exceptions after your specific exceptions because the program will apply the general exceptions to the code after it. This will generate an error because the program will not be able to apply the specific exceptions that are in the code after the general exceptions.*

*It also helps with the design and structure aspect of the program. When trouble shooting, the user can start from the general exceptions and work there way up to the specific exceptions.*

3. (15 points) When is it more correct to use exceptions, and when is it more correct to rely on if-else statements or other techniques?

*It is more correct to throw an exception when caller needs to handle an error. When the method can handle an exceptional code then it is more correct to use an if- else statement.*

4. (15 points) Why is it ideal for the File class to only deal with accessing and modifying file properties while the PrintWriter and Scanner classes deal with reading and writing file contents?

*The File class is best used to access and modify file properties because it does not contain methods to read or write file contents. The PrintWriter is best used to write file contents because this class has objects that contain methods to create a file and write data to a text file. The Scanner class has objects that contain methods used for reading file contents.*

5. (15 points) Write a short program used to read a set of floating-point numbers from a text file, add the numbers together, and print the total of these numbers to the console.

import java.io.File;  
import java.io.IOException;  
import java.util.Scanner;  
  
  
public class NumberTotals {  
 public static void main(String[] args) throws IOException {  
 Scanner numberFile = new Scanner(new File("Numbers.txt"));  
 String line1;  
 String line2;  
  
 while (numberFile.hasNext()) {  
 line1 = numberFile.nextLine();  
 line2 = numberFile.nextLine();  
 System.*out*.println(numberFile.nextLine());  
  
 }  
 }  
}