***CSC 3020***

***Java Programming***

**Assignment 08**

**40 points**

**Due 12/06/2023 (11:45 A.M.)**

Assignment Objectives:

* To get an overview of exceptions and exception handling.
* To declare exceptions in a method header.
* To throw exceptions in a method.
* To write a **try-catch** block to handle exceptions.
* To explain how an exception is propagated.
* To obtain information from an exception object.
* To develop applications with exception handling.
* To use the **finally** clause in a **try-catch** block.
* To rethrow exceptions in a **catch** block.
* To define custom exception classes.
* To write data to a file using the **PrintWriter** class.
* To read data from a file using the **Scanner** class.

**Solution to this assignment will not be posted on Canvas; however, any question can be discussed in the class upon request of a student.**

All assignments must be submitted by the Canvas. **No email or hard copy** is accepted. You must follow the following format:

1. For non-programming questions, use a word file to type your answers. Don’t use the text box on the Canvas to answer the questions or to write comments, we will not read it.
2. State your answer clearly.
3. For programming questions, include only the source file for each problem.
4. Submit your file to the Canvas. You must submit your assignment on time; otherwise, you will receive zero. In addition, you cannot submit your file more than one time.
5. There will be several folders on the Canvas. You need to upload your file(s) using the correct folder on the Canvas.
6. Name each file: “Assignment Number(Question number(s))”.
7. To upload your file(s):

* In Course Navigation, click the Assignments link.
* Click the title of the assignment.
* Click the **Submit** Assignment button.
* Add **File**. ...
* Add Another **File**. ...
* **Submit** Assignment. ...
* View **Submission**.

**It is your responsibility to make sure that each file is uploaded correctly. If you uploaded a wrong file, you receive zero; files will not be accepted after due date even if you have a prove that the file is created before the due date.**

**Make sure you review the Cheating & Plagiarism policy on Canvas.**

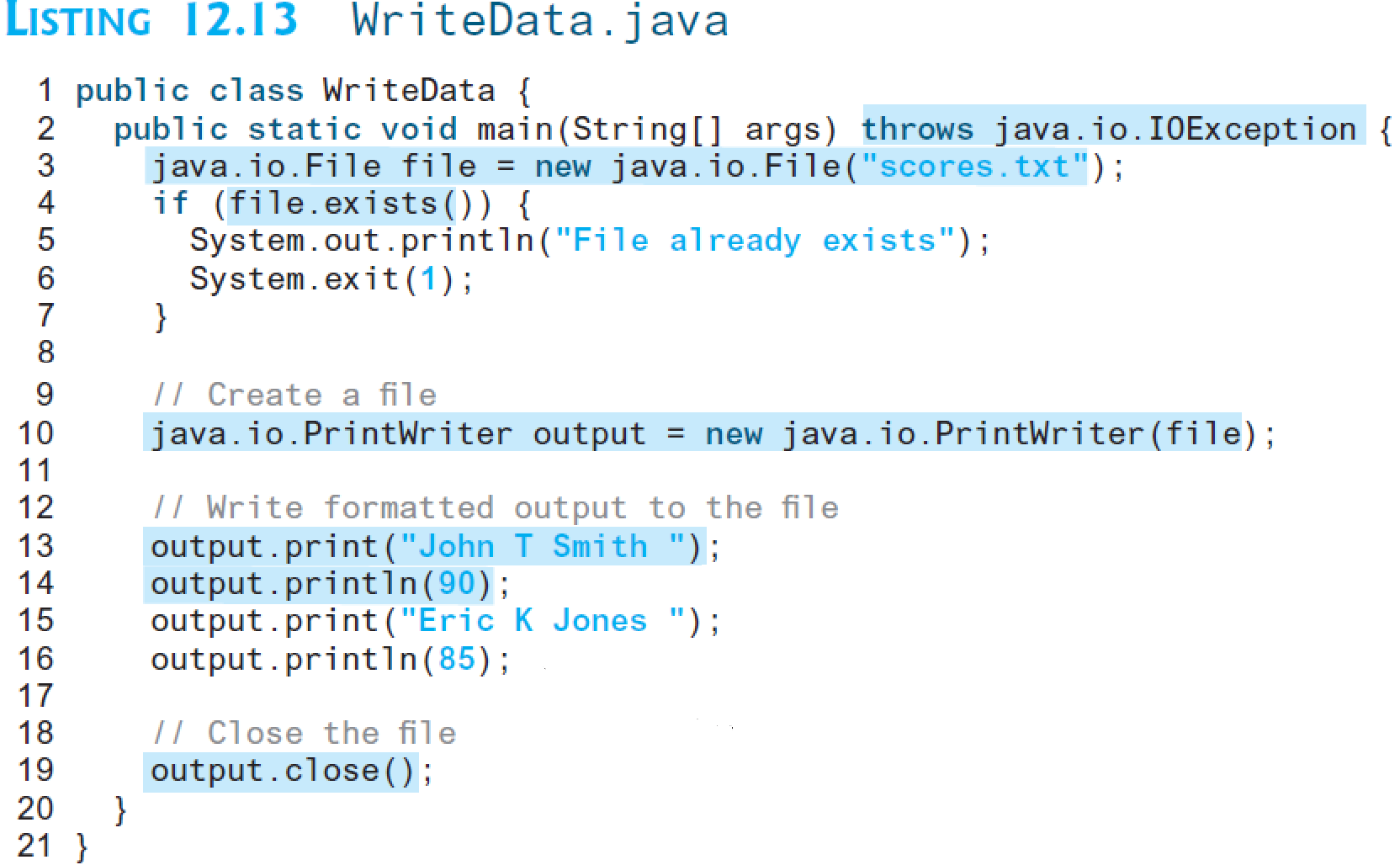
**Answer questions 1 to 2 in a separate single .txt file; include only your answers with questions numbers. Write a program for each of Q.3 - Q.6; save each program in a .txt file. Submit total of 5 .txt files by the due date.**

**Question 01 (12 Points)**

1. What is the purpose of declaring exceptions? How do you declare an exception in a method, and where? Can you declare multiple exceptions in a method header?  
   The purpose of declaring exceptions is to deal with runtime errors or conditions that occur during a program that make the program not perform as intended. Exceptions are declared using the ‘throw’ keyword which follows the parameters in the declaration of a method. You can declare multiple exceptions using commas.
2. What is a checked exception, and what is an unchecked exception?  
   A checked exception is an exception that the compiler mandates to be handeled either through it being caught in a try-catch block or declared in the header’s throws keyword.  
     
   Unchecked exceptions are not required to be handled explicitly – which is much less annoying.
3. How do you throw an exception? Can you throw multiple exceptions in one throw statement?  
     
   An exception is thrown using the ‘throw’ keyword. The word that follows after the ‘throw’ keyword is the name of the exceptions. Multiple exceptions cannot be thrown with just one throw statement – you have to use multiple throw statements.
4. What is wrong about creating a File object using the following statement?

**new** File("c:\book\test.dat");  
Backslashes are escape characters. :c

1. Can you use the File class for file I/O? Does creating a File object create a file on the disk?  
   The file class does not explicitly include ways to input and output to file, sadly you must use classes such as FileWriter to do that instead. Similarly creating a file object does not create a file on disk,
2. How do you create a PrintWriter to write data to a file? What is the reason to declare throws Exception in the main method in Listing 12.13, WriteData.java? What would happen if the close() method were not invoked in Listing 12.13?  
   You create a printwriter using the filewriter as follows:  
   FileWriter myfilewriter = new FileWriter(“lmao.txt”, false);  
   PrintWriter myprintwriter = new PrintWriter (myfilewriter);  
   //congrats, you can now use your print writer to print to file….I guess that’s useful?  
     
   The throws exception is there in case something goes wrong while it is writing.   
     
   Not closing a file that lead to funny situations such as accidentally writing to the wrong file and fun errors such as resource leaks. To avoid these, just close the file – or just use one file and hope nothing goes wrong (this is not a good plan, close the file).



1. How do you create a Scanner to read data from a file? What is the reason to define throws Exception in the main method in Listing 12.15, ReadData.java? What would happen if the close() method were not invoked in Listing 12.15?  
   A scanner can read data from a file if a file object is created, then a scanner objected is created that uses that file object as its parameter. It looks like this:  
   File myfile = new File(“ayylmao.txt);  
   Scanner scanner = new Scanner(myfile);  
   //congrats, you can now use your scanner to read the file. Neato.

The reason for the throws exception is because there is a checked exception when reading from files. It has to be there or else it won’t compile, it is annoying like that.  
  
Without using close() then the file does not close, this can lead to resource leaks, and making the file inaccessible because it is still technically ‘open’ by the program.



1. What will happen if you attempt to create a Scanner for a nonexistent file? What will happen if you attempt to create a PrintWriter for an existing file?  
   If there is no file for a scanner, the scanner will throw a File Not Found exception. This keeps happening to me when I mistype my file name in java.   
     
   If there is already an existing file for a PrintWriter, then the existing files content will be overrwritten by the printwriter.

**Question 02 (6 Points)**

Suppose you run the following code:

**public** **static** **void** main(String[] args) {

**try** {

m();

statement7;

}

**catch** (Exception2 ex {

statement8;

}

}

**public** **static** **void** m() {

**try** {

statement1;

statement2;

statement3;

}

**catch** (Exception1 ex1) {

statement4;

}

**finally** {

statement5;

}

statement6;

}

answer the questions:   
1. If no exception occurs, which statements are executed?   
Statement m is called, which tries 1,2,3 within the try block. The finally block is then executed, throwing statement 5, the block then ends, excecuting statement 6, then statement 7 is executed because m() has been executed.   
  
2. If statement2 throws an exception of type Exception1, which statements are executed?   
m() is executed, statement 1 is executed, statement 2 attempts to execute but throws an exception 1, leading to the catch block for exception 1 where statement 4 is executed. Statement 5 is then executed by by the finally block. Due to the exception occuring within m() statement 7 is skipped.   
  
  
3. If statement2 throws an exception of type Exception2, which statements are executed?   
m() is executed, leading to statement 1 being executed, and statement 2 attempting to be executed, however the exception occurs. The finally block is then triggered, leading to statement 5 being executed. Due to the exception 2 being called, statement 8 is executed by the catch block in main.   
  
4. If statement2 throws an exception that is neither Exception1 nor Exception 2, which statements are executed?

**m() is called, leading to statement 1 being executed, statement 2 attempts to be executed, but throws an error. The finally block runs, running statement 5.**

**Question 03 (5 Points)**

Write a program that demonstrates how various exceptions are caught with catch (Exception exception). Define classes ExceptionA (which inherits from class Exception) and ExceptionB (which inherits from class ExceptionA). In your program, create try blocks that throw exceptions of types ExceptionA, ExceptionB, NullPointerException and IOException. All exceptions should be caught with catch blocks specifying type Exception.

**Question 04 (3 Points)**

Write a program demonstrating that the order of catch blocks is important. If you try to catch a superclass exception type before a subclass type, the compiler should generate errors.

**Question 05 (8 Points)**

Twenty students were asked to rate on a scale of 1 to 5 the quality of the food in the student cafeteria, with 1 being “awful” and 5 being “excellent.” The 20 responses are stored in “numbers.txt” file. Read the responses from “numbers.txt” file, determine the frequency of each rating and output it to the text file “output.txt”.

The responses should be read from the file by using a Scanner. Use method nextInt to input one integer at a time from the file. The program should continue to read responses until it reaches the end of the file. The results should be output to the text file “output.txt”. Use a PrintWriter to create a file called “output.txt”. Each integer should be written using method printf.

Use FileNotFoundException and NoSuchElementException Exceptions.

Sample input and output are shown in the below table.

|  |  |
| --- | --- |
| **numbers.txt**  *3 1 2 5 3 3 2 2 3 3 2 1 4 3 1 4 4 5 3 3* | **output.txt**  *Rating Frequency*  *1 3*  *2 4*  *3 8*  *4 3*  *5 2* |

**Question 06 (6 Points)**

Write a program that replaces all the occurrences of a specified string with another specified string from a text file. Your program should get the arguments from the command line.

For example, invoking

**java javaFile apples lemons filename**

replaces the string **apples with lemons** from the specified file.

Check if the user entered correct number of arguments in the command line and if the file exist.

Sample input and output of the same file is shown below:

|  |  |
| --- | --- |
| **fruits.txt**  *two apples + one orange = two apples*  *one orange + three apples = three apples*  *one grapefruit + two pomegranates = zero apples* | **fruits.txt**  *two lemons + one orange = two lemons*  *one orange + three lemons = three lemons*  *one grapefruit + two pomegranates = zero lemons* |