CS-207: Programming II
Spring 2016
Northeastern Illinois University
Homework #8: Due 04/07/16 at 9:00 a.m.
Exception Handling

Problem #1

Create a class named `StringParser` that has the following:

- A public static method named `findIntegerDivisors` that takes a `String` and two `char` variables as parameters (in that order) and does not return anything.
- The method should find the integer value that is located in between the two characters.
- The method should then print out all of the numbers between 1 and the integer (inclusive) that divide evenly into the integer on the same line separated by spaces.
- You **cannot** assume that only valid integers or nothing will appear in between the specified characters. If the value is not valid or there is nothing in between the specified characters, print out `"Invalid characters"` followed by calling the `toString` method of the exception object that is thrown.
- Make sure to handle the most specific exception class (i.e. do **NOT** use the superclass `Exception` to catch the exception object).
- You can assume that the second `char` parameter will always follow the first `char` parameter. However, you **cannot** assume that the parameters will be different from each other.
- You may only use one loop (for finding the divisors).
- You may not use any conditional statements (including switch statements and the tertiary operator) other than a single if-statement for finding the divisors.
- Download the `StringParserTest` from the NeededFiles.zip file and compile and run it.
- If you created your class and method correctly, you will see the output below.
- Place your StringParser file into the Homework8 folder to be submitted to D2L.

```
String is: rugtsbckgus!32*
1 2 4 8 16 32

String is: disdkfjs<873>sfjsldkfiwx
1 3 9 97 291 873

String is: rujfbgl&%fkslga
Invalid characters
java.lang.NumberFormatException: For input string: ""

String is: rusbdi#1038#jjdksu
1 2 3 6 173 346 519 1038

String is: rusbdi^10.38^jjdksu
Invalid characters
java.lang.NumberFormatException: For input string: "10.38"
```

Problem #2

Given a binary number as a `String`, find its 1's complement. The 1's complement of a binary number is another binary number created by toggling all the bits in it. In other words, change the 0s to 1s and the 1s to 0s.

Create a class named `Complement` that has the following:

- A public static method named `onesComplement` that takes a `String` as a parameter and returns a `String` that is the 1's complement of the parameter.
- If the `String` is not a valid binary number, the method should throw an `InputMismatchException` that has the `message`: `"Not a valid binary number"`. Don't forget that you need an import statement to use the `InputMismatchException` class (see Lecture 13)!
- You may not use any loops.
- You may use at most one conditional statement, but it cannot include any else if or else blocks.
- Download the `ComplementTest.java` file from the NeededFiles.zip file. Note that it has an empty method named `handleComplementCall` that takes one parameter, a `String`, and does not return anything.
- The `handleComplementCall` method should `try` to call the `onesComplement` method from the `Complement` class and pass in the parameter `s` to the method.
- If `onesComplement` throws an exception, you should catch it and print out the result from calling the `printStackTrace` method. You may not use any loops or conditionals (use try-catch!!).
- Make sure to handle the most specific exception class (i.e. do **NOT** use the superclass `Exception` to catch the exception object).
- Run the `ComplementTest` class. If you created your class and method correctly, you will see the output below. Note that the line numbers for the stack trace may be slightly different in your output than in the sample output.

- Place the `Complement` and `ComplementTest` files into the Homework8 folder to be submitted to D2L.

```
010101
111
00000000
java.util.InputMismatchException: Not a binary number
    at Complement.onesComplement(Complement.java:7)
    at ComplementTest.handleComplementCall(ComplementTest.java:17)
    at ComplementTest.main(ComplementTest.java:10)
```

Problem #3

Download the `scores.txt` file from the NeededFiles.zip file. Note that there should be multiple lines in the file, where each line has a first and last name followed by 3 numeric values. Create a class named `FindAverages` that does the following (in the `main` method or another method - your choice):

- Open the file and read from it.
- Calculate the average score for each individual (i.e. each line) in the file.
- Print out the first and last name followed by the individual's average score. Your output should match the output below.
- Do not hard-code any values (i.e. I don't want to see 10 `nextLine` statements!).
- Make sure to use the correct imports!
- If the file cannot be found, print out `"File, where art thou?"`
- You may not use any conditional statements (including switch statements and the tertiary operator).
- Place the `FindAverages` file into the Homework8 folder to be submitted to D2L.

```
John Smith, Average: 84.26666666666667
Cherry Pie, Average: 48.166666666666664
Banana Split, Average: 92.03333333333335
Yoko Ono, Average: 47.73333333333333
Happy Gilmore, Average: 77.2
Sam Adams, Average: 63.6
Pat Mann, Average: 173.4333333333333
Oliva Munn, Average: 129.90333333333334
Bernie Sanders, Average: 33.333333333333336
Winston Churchill, Average: 5.5
```

A note on cheating/plagiarism:
A plagiarism detector is used on all submitted code (across all sections) for homework assignments. If the plagiarism detector determines that 25% or more of your code for a particular assignment is plagiarized, you will receive a zero (i.e. an F) for that homework assignment, regardless of whether you

cheated from someone or vice-versa. If you plagiarize half or more of the total homework assignments, you will receive a zero for the entire homework percentage.

Submitting your assignment to D2L

1. Make sure your name and assignment number are in the .java file(s) (as comments) and text file.

2. Place all your files in a folder and compress (i.e. .zip) the folder. Submit the .zip file to the Homework #8 folder on D2L. You should submit only one file - the .zip file. Do **NOT** upload multiple files.

3. Turn your homework in to D2L by the specified deadline (no late homework will be accepted - see syllabus for policies)