**CMSC 201 Introduction to Java**

**Fall 2017**

**Project # 9**

**Concepts Tested in this Program**:

* Error Handling
* Text I/O

**Project 9 consists of three small programs:**

**Program 9\_1:**

Write a program that prompts the user to read two [integers](https://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?)  and displays their sum . If anything but an [integer](https://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?)  is passed as input, your program should catch the InputMismatchException that is thrown and prompt the user to input another number by printing "Please enter an [integer](https://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?) ."

Name your program’s file as InputMismatch.java

### SAMPLE RUN #1: java InputMismatch

Highlight:  Show Highlighted Only  

Enter·an·integer:2.5·↵

Please·enter·an·integer.↵

Enter·an·integer:4.6·↵

Please·enter·an·integer.↵

Enter·an·integer:hello!·↵

Please·enter·an·integer.↵

Enter·an·integer:7·↵

Enter·an·integer:5.6·↵

Please·enter·an·integer.↵

Enter·an·integer:9.4·↵

Please·enter·an·integer.↵

Enter·an·integer:10↵

17↵

**Program 9\_2:**

Write the bin2Dec (String  binaryString) method  to convert a binary string  into a decimal number (your method  should return an integer ). Implement the bin2Dec method to throw a NumberFormatException if the string is not a binary string.   
  
Test your method in a [program](https://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?)  that prompts the user to input a binary string and uses your bin2Dec method to print out the decimal equivalent of that string. If the method throws an error, have your program catch it and print "Not a binary string." before terminating.

Name your program’s file as NumberFormat.java

**SAMPLE RUN #1:** java NumberFormat

Enter·a·binary·string:1011000101↵ 709↵

**Program 9\_3:**

Write a [program](https://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?)  that reads lines from a file and prints them out, removing all occurrences of a specified [string](https://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?)  from the lines.  
  
For example, if the [class](https://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?)  that housed your [program](https://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?)  was called RmvText, running this prompt on the command line:  
  
java RmvText John filename  
  
Would print out the contents of the file filename without any instances of the [string](https://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?)  John.   
  
Your [program](https://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?)  should get the [arguments](https://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?)  from the command line.

Name your program’s file as RmvText.java

### SAMPLE RUN #1: java RmvText going input

I'm·to·the·market.↵

How's·it?↵

Things·are·great.↵

You'd·better·get!↵

**Project 9 Submission requirements:**

**Test Plan**

|  |  |  |  |
| --- | --- | --- | --- |
| **Cases** | **Input** | **Expected Result** | **Actual Result** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Deliverables:**

* Java files (source code)
* Word document should include:
* Test Plan, screen snapshots of outputs based on each Test Case
* Algorithm, Flowcharts, UML
* Lessons Learned
* Check List

**Deliverable format**: The above deliverables should be packaged in two compressed files in the following format.

1. FirstInitialLastName\_Project8\_doc.zip [a compressed file containing the following]

                             FirstInitialLastNamePr9\_1.docx

FirstInitialLastNamePr9\_2.docx

FirstInitialLastNamePr9\_3.docx

1. FirstInitialLastName\_Project9\_Moss.zip [a compressed file containing only the following]

Your completed assignment should be submitted on **Blackboard ->Course Content-> Week xx->"Project 9"** no later than the assigned due date. You should include one block comment at the top of each program containing the course name, the project number, your name, the date and platform/compiler that you used to develop the project, for example

/\*

 \* Class: CMSC201

 \* Instructor:

 \* Description: (Give a brief description for each Program)

 \* Due: 11/12/2017

 \* I pledge that I have completed the programming assignment independently.

   I have not copied the code from a student or any source.

   I have not given my code to any student.

   Print your Name here: \_\_\_\_\_\_\_\_\_\_

\*/

**Check List**

|  |  |  |  |
| --- | --- | --- | --- |
| **#** |  | **Y/N** | **Comments** |
|  | **Source java files** |  |  |
|  | **Compressed files:** |  |  |
|  | FirstInitialLastName\_Project9\_Moss.zip |  |  |
|  |  |  |  |
|  | FirstInitialLastName\_Project9\_doc.zip |  |  |
|  | **Program compiles** |  |  |
|  | **Program runs** |  |  |
|  | **Checklist is completed and included in the Documentation** |  |  |
|  | **Documentation file:** |  |  |
|  | **Comprehensive Test Plan** |  |  |
|  | **Screenshots based on Test Plan** |  |  |
|  | **UML Diagram** |  |  |
|  | **Algorithms/Pseudocode** |  |  |
|  | **Flowchart** |  |  |
|  | **Lessons Learned** |  |  |

**Notes:**

* Proper naming conventions: All constants, except 0 and 1, should be named. Constant names should be all uppercase, variable names should begin in lower case, but subsequent words should be in title case. Variable and method names should be descriptive of the role of the variable or method. Single letter names should be avoided.
* Documentation: The documentation requirement for all programming projects is one block comment at the top of the program containing the course name, the project number, your name, the date and platform/compiler that you used to develop the project. If you use any code or specific algorithms that you did not create, a reference to its source should be made in the appropriate comment block. Additional comments should be provided as necessary to clarify the program.

Indentation: It must be consistent throughout the program and must reflect the control structure. For more examples on documentation please refer to the online template.

**Grading Rubric**

**CMSC 201 Project 9**

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**PROGRAMMING**

Doesn’t Compile -100 pts \_\_\_\_\_ (second submission is allowed -50 pts will be deducted for 2nd attempt)

All requirements are implemented and program compiles

Accuracy:

Doesn’t Pass private instructor tests -10 pts \_\_\_\_\_

Execution: runs with errors (either run-time or logic errors) -40 pts \_\_\_\_\_

**Documentation:**

Pseudo-code (Algorithm) was not provided - 5 pts \_\_\_\_\_

UML class Diagram was not provided - 5 pts \_\_\_\_\_

Flowchart was not provided - 5 pts \_\_\_\_\_

Table of test cases (Test Plan) was not provided – 5 pts\_\_\_\_\_

Screen shots of IDE (or console) run was not provided or not related to a Test Plan - 5 pts \_

Documentation within source code was missing or incorrect - 5 pts \_\_\_\_\_

Lessons Learned were not provided - 5 pts \_\_\_\_\_

Check List was not provided – 5pts

**Programming Style:**

Incorrect use of indentation, statements, structures - 5 pts \_\_\_\_\_

User interface

Not clear to user how data is to be entered; UI does not follow requirements - 10 pts \_\_\_\_\_

Output is easy to understand - 4 pts \_\_\_\_\_