CGS 3767 Operating Systems - project 1 : cgs3767pgm1a

Professor: Michael Robinson
e-mail   : michael.robinson@fiu.edu
Web Page : www.cs.fiu.edu/~mrobi002/teaching

a - Install virtualbox in your buffalo partition at FIU
b - Install Fedora as a virtual machine inside virtualbox
c - Install Java JDK inside Fedora
d - Create a folder called projects inside Fedora
e - Create a folder called backups inside Fedora
f - Using Fedora's terminal (equivalent to MSDOS in Windows)
    create the java program described below
g - Copy and rename your files in the projects folder to
    your backup folder
h - Do above steps d,e,f and g in Windows using the cmd (MSDOS)
    line as shown in class.

- Program must be named: yourLastNameFirstLetterOfYourFirstNamepgm1.java
- Turn in the signed source code on paper, and email me the source code.
- Make sure the program is properly documented and aligned uniformally,
  looking professionally, I will take points off if it not.
- Include the following header in every program:

```
/*****************************************************************
Author     : Your Name
Course     : CGS 3767 days and time
Professor  : Michael Robinson
Program #  : Program Purpose/Description
             {A brief description of the program }

Due Date   : MM/DD/YYYY

Certification:
I hereby certify that this work is my own and none of it is the work of any other person.

 ..........{ your signature }..........
*****************************************************************/
```

Purpose of this project:

- Create a very simple Java program in Windows and Fedora using their Editors
  *** DO NOT USE ANY IDE ***
- Windows : From the command line (MSDOS), use the edit or notepad editor
- Fedora : From the Terminal use the gedit editor
- Compile and run this program using the MSDOS/Terminal
```
      javac robinsonMpgm1.java
      java  robinsonMpgm1.class
```
- Use multiple variables of Primitive Data Types and the String Class by
  declaring them, and assigning values to them in the main method.
- Create some methods that accept, and others do not accept parameters.
  (MAKE SURE THE METHODS NAMES DESCRIBE WHAT THEY DO example: addNumbers)
- Inside the methods do calculations when needed and print results.
- Use print, println, and printf.
- Use \n and \t
- Use remarks to document your program.
- Use for loops

How:
  1 - In the main method, using the proper Primitive Data Types,
      - Create the following variables with the following values:

```
   Data Type    Variable Name      Variable Contents
     ???        myName          =  place your name here
     ???        creditsTaken    =  place your credits taken this semester
     ???        totalCredits    =  place your total amount of credits taken
     ???        GPA             =  place your current GPA
     ???        major           =  place your major
     ???        className       =  place the name of this class
```

- Call a method named myInfo PASSING the previous variables

- Create the following variable with the following value:

```
   Data Type    Variable Name      Variable Contents
     ???        maxValue        =  100
```

- Call the following methods PASSING the maxValue variable
  - addNumbers
  - substractNumbers
  - multiplyNumbers
  - divideNumbers
  - modNumbers

2 - Create the following methods ACCEPTING their corresponding data variables
  - myInfo
  - addNumbers
  - substractNumbers
  - multiplyNumbers
  - divideNumbers
  - modNumbers

3 - In the myInfo method, using the System.out.printf and \n commands print the information send from the main method and received by this method e.i.

```
   Hi my name is ..,
   my major is ..,
   I have completed .. credits,
   I am taking .. credits,
   This class's name is ..
```

- In the addNumbers method, using the System.out.print and \n commands print the following computations: (make sure your program does the computations)

```
maxValue + 1 = ??
maxValue + 2 = ??
maxValue + 3 = ??
maxValue + 4 = ??
maxValue + 5 = ??
```

- In the substractNumbers method, using the System.out.print and \t commands print the following computations: (make sure your program does the computations)

```
maxValue - 1 = ??
maxValue - 2 = ??
maxValue - 3 = ??
maxValue - 4 = ??
maxValue - 5 = ??
```

- In the multiplyNumbers method, using the System.out.println command print the following computations: (make sure your program does the computations)

```
maxValue * 1 = ??
maxValue * 2 = ??
maxValue * 3 = ??
maxValue * 4 = ??
maxValue * 5 = ??
```

- In the divideNumbers method, using the System.out.printf command ONLY print the following computations: (make sure your program does the computations)

```
maxValue / 1 = ??
maxValue / 2 = ??
```

```
                    maxValue / 3 = ??
                    maxValue / 4 = ??
                    maxValue / 5 = ??

   - In the modNumbers method, using the System.out.print command ONLY print the following
     computations:  ( make sure your program does the computations )

                    maxValue % 1 = ??
                    maxValue % 2 = ??
                    maxValue % 3 = ??
                    maxValue % 4 = ??
                    maxValue % 5 = ??

4 - From the main method call a method named sumOfDigits(), without passing any parameters
  - Create a method named sumOfDigits()
  - In the sumOfDigits() method declare the variable N of type int
  - Assign the value 100 to the variable N.
  - Using the sum of digits formula: (1 + N)*(N/2)
    print the total amount of the sum of digits from 1 to 100

    The formula (1 + N)*(N/2) will calculate
    the sum of all the numbers from 1 to 100 (1 + 2 + 3 + 4 + . . + 100),
    The formula is: (1 plus N) times (N divided by 2).

5 - From the main method call a method named forLoop(), without passing any parameters
  - Create a method named forLoop()
  - In the forLoop() method declare the variable total of type int
  - Using a for loop to the variable total, add the sum of all numbers from 1 to 100
  - Use the System.out.printf and \n commands to print the variable total
```