```
COP-2250 - Java Programming
Professor: Michael Robinson
e-mail   : mrobi002@cs.fiu.edu
Web Page : www.cs.fiu.edu/~mrobi002/teaching

Program 2c

Purpose: Implement methods, if, else if, else, while, for, print, println, printf, Scanner,
         primitive types, String class with several methods, single dimension array, modulus,
         error trapping.

Turn in the source code on paper, and email me the source code, the input and output files.
Make sure the program is properly documented and aligned uniformally, looking professionally,
I will take points off if it not.

In addition to the main() method, create methods for each of the following:
 !!! make sure each method's name describes what it does. e.i. addNumbers(); !!!

   ***************************************************************************
   * EVERY EXERCISE MUST THE DONE USING METHODS.                             *
   * make sure each method's name describes what it does. i.e. addNumbers(); *
   ***************************************************************************

1 - Implement division by 0, with error trapping, using if and while() commands, make sure to use
    "casting" e.i. float result = (float)int/int;
    How: Using a while loop, read 2 numbers from the user.
         Using the if statement, test that the second number in not zero, if it is zero inform
         the user of the error, and do the loop again to ask for a correct second number.
         if the second number is NOT a zero, do the division, display all numbers and the
         computation using labels, the result MUST have 2 decimal places,
             e.i. "The first number 10 divided by the second number 5 is 2.00"
         then exit the while loop entering the value 999 for the first or the second number.

2 - Using the loop of your choice, and the print of your choice, display all numbers from 0 to 1,000
    where  "mod 5 == 3".  Hint: %


************************************************************************************************
* NOTE: Every method dealing with your name, MUST be called from main, PASSING your name to it.*
************************************************************************************************

3 - At the main method, create a single dimension array containing your
    "first name $ middle name % last name and ONE space"
    i.e. "George & Washington % Rodriguez "

    NOTE: Every method dealing with your name, MUST be called from main, PASSING your name to it.

    Using the for loop, if, else if, else, ||, and &&, printf, and any other command you want,
    examine each character of your name and determine if it is:
    a space, a vowel (a,e,i,o,u), a consonant (b,w,g, etc) the symbol % or the symbol $,
    and using the current value of x in your for loop print:

    "character [x] located at position x is a consonant" or
    "character [x] located at position x is a vowel" or
    "character [x] located at position x is a space" or
    "character [x] located at position x is a not a vowel or a consonant or a space".

     ie:  My name is [George $ Washington % Rodriguez]
          character [G] located at position 0 is a consonant
          character [e] located at position 1 is a vowel
          character [o] located at position 2 is a vowel
          character [r] located at position 3 is a consonant
          character [g] located at position 4 is a consonant
          character [e] located at position 5 is a vowel
          character [ ] located at position 6 is a space
          character [$] located at position 7 is not a vowel or a consonant or a space
          character [ ] located at position 8 is a space
```

```
        character [W] located at position 9 is a consonant
        character [a] located at position 10 is a vowel
        character [s] located at position 11 is a consonant
        character [h] located at position 12 is a consonant
        character [i] located at position 13 is a vowel
        character [n] located at position 14 is a consonant
        character [g] located at position 15 is a consonant
        character [t] located at position 16 is a consonant
        character [o] located at position 17 is a vowel
        character [n] located at position 18 is a consonant
        character [ ] located at position 19 is a space
        character [%] located at position 20 is not a vowel or a consonant or a space
        character [ ] located at position 21 is a space
        character [R] located at position 22 is a consonant
        character [o] located at position 23 is a vowel
        character [d] located at position 24 is a consonant
        character [r] located at position 25 is a consonant
        character [i] located at position 26 is a vowel
        character [g] located at position 27 is a consonant
        character [u] located at position 28 is a vowel
        character [e] located at position 29 is a vowel
        character [z] located at position 30 is a consonant
        character [ ] located at position 31 is a space
```

4 - Using a for loop, display to the screen, the string containing your name so that each
loop will NOT contain the last character on the previous line,
with the length of the string being printed, and the string surrounded by square brackes []

NOTE: Every method dealing with your name, MUST be called from main, PASSING your name to it.

```
    i.e:  32 [George $ Washington % Rodriguez ]
          31 [George $ Washington % Rodriguez]
          30 [George $ Washington % Rodrigue]
          29 [George $ Washington % Rodrigu]
          28 [George $ Washington % Rodrig]
          27 [George $ Washington % Rodri]
          26 [George $ Washington % Rodr]
          25 [George $ Washington % Rod]
          24 [George $ Washington % Ro]
          23 [George $ Washington % R]
          22 [George $ Washington % ]
          21 [George $ Washington %]
          20 [George $ Washington ]
          19 [George $ Washington]
          18 [George $ Washingto]
          17 [George $ Washingt]
          16 [George $ Washing]
          15 [George $ Washin]
          14 [George $ Washi]
          13 [George $ Wash]
          12 [George $ Was]
          11 [George $ Wa]
          10 [George $ W]
          9 [George $ ]
          8 [George $]
          7 [George ]
          6 [George]
          5 [Georg]
          4 [Geor]
          3 [Geo]
          2 [Ge]
          1 [G]
```

NOTE: Every method dealing with your name, MUST be called from main, PASSING your name to it.

5 - Print your name after changing it to all upper case letters, and also to lower case.

6 - Print your LAST name ONLY extracting it from your full name, using the substring method in the String class.

7 - Print your name after taking all spaces out.

8 - Print your name after making all vowels upper case, and all consonants lower case.

9 - Using a for loop, print your name backwards.

A - Print your name in ASCII values.

Turn in the source code on paper, and email me the source code.
Make sure the program is properly documented and aligned uniformally, looking professionally.