Assignment 5

Object Oriented Programming

March 8, 2021

For each of the following problems please upload <filename>.py file, where filename is specified in the problem description. Upload your solutions to Canvas before the assignment deadline. You can also zip all the files into a single file and upload that instead. You can continue to resubmit until the deadline.

1. **Pizza Party**
   You are arranging a pizza party for all of your friends. You want to figure out how much pizza you can order without exceeding your budget. Write a program that accepts the number of people attending your party, the budget for your party, the price per slice of pizza, and the price per pie of pizza. Your program should then identify how many slices each person wants and determine how many individual slices and pies to order without exceeding your budget. If you exceed your budget, the program should warn the user they can't place their order. Note that for the purpose of this program a pizza pie contains 8 slices. And note that the restaurant you are ordering from will not sell you more than 7 individual slices at a time (if you need to purchase 8 slices you will have to buy a whole pie).

   You do not need to validate the first four input statements (total budget, cost per slice, cost per pie and # of people attending party). You will have to validate the user's input when prompting them for the number of slices for each person coming to the party (i.e. Enter number of slices for person #1).

   Sample Program:

```
Enter budget for your party: 100
Cost per slice of pizza: 2.50
Cost per whole pizza pie (8 slices): 12.50
How many people will be attending your party? 10
Enter number of slices for person #1: -2
Not a valid entry, try again!

Enter number of slices for person #1: 2
Enter number of slices for person #2: 3
Enter number of slices for person #3: 4
Enter number of slices for person #4: 2
Enter number of slices for person #5: 3
Enter number of slices for person #6: 4
Enter number of slices for person #7: 2
Enter number of slices for person #8: 1
Enter number of slices for person #9: 5
Enter number of slices for person #10: 4
You should purchase 3 pies and 6 slices
Your total cost will be: 52.50
You will still have 47.50 left after your order
```

```
Enter budget for your party: 10
Cost per slice of pizza: 2.50
Cost per whole pizza pie (8 slices): 12.50
How many people will be attending your party? 4
Enter number of slices for person #1: 2
Enter number of slices for person #2: 2
Enter number of slices for person #3: 2
Enter number of slices for person #4: 2
Your order cannot be completed.
You would need to purchase 1 pies and 0 slices
This would put you over budget by 2.50
```

```
Enter budget for your party: 10
Cost per slice of pizza: 2.50
Cost per whole pizza pie (8 slices): 12.50
How many people will be attending your party? 4
Enter number of slices for person #1: 1
Enter number of slices for person #2: 1
Enter number of slices for person #3: 1
Enter number of slices for person #4: 1
You should purchase 0 pies and 4 slices
Your total cost will be: 10.00
You will have no money left after your order.
```

This program should be named as follows: $LastName\_FirstName\_assignment5\_problem1.py$.
For example, $Haroon\_Shaheer\_assignment5\_problem1.py$.

2. **Dynamic Gradebook**
   Write a gradebook program that lets a teacher keep track of test averages for his or her students. Your program should begin by asking the teacher for a number of students in their class as well as the total # of tests that will be given to the class. Validate this information to ensure that the numbers entered are positive.

   Next, prompt the teacher to enter in scores for each student. Ensure that the values entered are positive - if they aren't you will need to re-prompt them. Hint: you may need to use nested loops here! A "while" loop can be placed inside of a "for" loop, if necessary.

   Once your program has collected all test scores for a student it should display that student's average and move onto the next student. When all students have been calculated the program should compute the overall average score for the entire class.

   Here's a sample running of your program:

```
How many students are in your class? -5
Invalid # of students, try again.

How many students are in your class? 3
How many tests in this class? -10
Invalid # of tests, try again.
How many tests in this class? 2

Here we go!

**** Student #1****
Enter score for test #1: -50
Invalid score, try again
Enter score for test #1: 50
Enter score for test #2: 75
Average score for student #1 is 62.50

**** Student #2****
Enter score for test #1: 100
Enter score for test #2: 90
Average score for student #2 is 95.00

**** Student #3****
Enter score for test #1: -10
Invalid score, try again
Enter score for test #1: -20
Invalid score, try again
Enter score for test #1: -30
Invalid score, try again
```

```
Enter score for test #1: 90
Enter score for test #2: 80
Average score for student #3 is 85.00

Average score for all students is: 80.83
```

This program should be named as follows: $LastName\_FirstName\_assignment5\_problem2.py$.
For example, $Haroon\_Shaheer\_assignment5\_problem2.py$.

3. **Prime Numbers**
   Recall term test #1, you had to write a program that determined whether a number was prime or not. For this problem, we would like to print all the prime numbers within a given range so that only 10 numbers print per line. Align the numbers so that they stack neatly on top of one another in all cases (i.e. the table should line up no matter what number range you are analyzing). Here's a sample running of the program:

```
Start number: 1
End number: 100
  2   3   5   7  11  13  17  19  23  29
 31  37  41  43  47  53  59  61  67  71
 73  79  83  89  97
```

```
Start number: 1
End number: 1000
   2    3    5    7   11   13   17   19   23   29
  31   37   41   43   47   53   59   61   67   71
  73   79   83   89   97  101  103  107  109  113
 127  131  137  139  149  151  157  163  167  173
 179  181  191  193  197  199  211  223  227  229
 233  239  241  251  257  263  269  271  277  281
 283  293  307  311  313  317  331  337  347  349
 353  359  367  373  379  383  389  397  401  409
 419  421  431  433  439  443  449  457  461  463
 467  479  487  491  499  503  509  521  523  541
 547  557  563  569  571  577  587  593  599  601
 607  613  617  619  631  641  643  647  653  659
 661  673  677  683  691  701  709  719  727  733
 739  743  751  757  761  769  773  787  797  809
 811  821  823  827  829  839  853  857  859  863
 877  881  883  887  907  911  919  929  937  941
 947  953  967  971  977  983  991  997
```

```
Start number: 9900
End number: 10100
  9901  9907  9923  9929  9931  9941  9949  9967  9973 10007
 10009 10037 10039 10061 10067 10069 10079 10091 10093 10099
```

Hints

- Figure out how to test if a number if prime. Then figure how to determine all the prime numbers in a range. Finally reason about how to display these numbers in the format requested.

This program should be named as follows: $LastName\_FirstName\_assignment5\_problem3.py$. For example, $Haroon\_Shaheer\_assignment5\_problem3.py$.