Intro to CS in Python
Math Homework

Intro

This homework is worth 90 points. There are three parts and each part is worth 30 points. All parts must be turned in to the instructor at the deadline decided by the instructor. Otherwise the assignment will be marked late.

General Notes: Please read this

Welcome to the Intro to Computer Science in Python homework material at MVCDS. This homework is the only homework that will have this section which outlines the requirements of homework assignments. Generally the homework assignments will be given out periodically. That is to say if you are given an assignment on Monday, you should turn in the assignment before the following Monday when the next assignment is given out. This may change on your instructor's request: if they are teaching this in workshops that meet four times a week the homework might be due sooner. If you are doing this on your own you can make your own deadline for your assignments, but please keep in mind that the assignments are made with a time constraint of a week or so in mind, so please adhere to that. The assignments lose their affect if done over a month.

How will grading work?

Each homework assignment will have multiple parts to it. Each section will have a description and there will be a general theme of knowledge that is built up during the homework assignment's parts. This will be a core aspect of every homework. Keeping this in mind, here are a few things that you will be graded on:

- Program correctness. This should be the top priority of your assignments for this collection of work. You will get the lion's share of your points from this requirement. However, that does not mean you will not gain or lose points based on the other requirements outlined on this homework. Your results must not only match the sample input given to you with the homework, but the text cases that you are graded on by the instructor. The slightest variation will result in points being lost.
  - **Note**: Small differences in the realm of spaces or frivolous characteristics compared to the purpose of the homework. If you are given a homework assignment that prints out information about local business and your section on the address of a business does not have the right formatting, it will lose points. However it will not lose as nearly as much points as in the case of **not** print the address at all, having wrong information, or the wrong business. We care about correct logic in the assignment more than how pretty it looks.
- The last submission that you submit before the acceptable due date is the one that will be graded. Please be aware of this because there is a zero tolerance for changing a homework after it is due and will result in it being counted as late.
- The code you submit will be looked at as well. Please have well-written code with meaningful names and correct program logic. Test your code with your own test cases, not only the ones we give you. You will lose points if your answers are correct but your logic is wrong; this will be explained in-depth during the course.
- After this homework, your "coding-style" will be examined as well. These elements include the organization of code, commenting what your code does, the naming of your variables, function usage, and general readability. You must develop good habits early on to aid in your development as a programmer.
- "Can I use logic not taught to me yet on this homework?" The short answer is no, but we will not take away points if you do. The long answer is that the homework assignments are designed to have a

specific purpose on what you should learn and what tools you should use to achieve the assignment's goals. The purpose of homework that make you think "there must be a better way!" or "this is very repetitive" is to have you ponder what could be used to fix this, appreciate the new tool to come, and to fully master the task before moving on to the next challenge. If you do not do this, you **will not lose points** but you will be encouraged to try the way outlined in the homework assignment. Eat your greens, it's good for you.

- You must submit a program file with a correct name in order for us to keep track of everything. Points will be taken if you do not have a file named in a manner similar to this:

```
hw1_part1.py
hw1_part2.py
hw1_part3.py
```

The following is also acceptable, if you are weird about sorting things like the writer of this assignment.

```
hw01_part1.py
```

Each will be submitted separately so they can be tested separately. Remember to submit them all before the deadline assigned by your instructor.

Late homework

Each homework assignment will have multiple parts. Some will have two, others will have four, but most will have three parts. If any part of your assignment is late, then all of them are late. Your instructor may decide for the course to allow late days. The usual rule for late days is that a student will be given **3 late days** at the start of the course. If you wish to use a late day, you must email your instructor that you are doing so for a homework assignment. This email must happen before the due date. When this email occurs, the instructor will reduce your late day count by 1. If you reach 0 late days, you will not be able to use anymore late days on an assignment. The usage of a late day means that if a homework assignment is due at midnight on Friday, it will be due the following day at Saturday at the same time.
If you believe you qualify for extra time for personal reasons of any kind, please provide valid proof from the Nurse's Office or the equivalent.

Wing IDE vs. our answers

Currently, we will ask you to submit the homework to the instructor via email to their email address. In the future, we will work on making an online submission process to check your homework. We will try to give you good test cases and to answer all the questions that you ask (within reason) to balance this drawback.

Homework description

Part 1: Custom Greeting Cards

In this part of the homework, you will make a greeting card with the default text below:

Hello <person>,

Intro to CS in Python
Math Homework

> **I have started learning Computer Science at <location> and I am really <emotion> to get started. After working on my <noun> for this class, I will go <verb> with my friends after school. I just got back from <noun> and I feel very <adjective>. I hope to <verb> my homework soon, that would make me feel <adjective>.**

Talk to you later, <different person>

Your program will ask the user using the function we learnt in class (raw_input) . For each input, you should ask for the specific type of word needed (verb, emotion, adjective, … etc). Once that is done you will take all their input and add it into where it is needed in the given letter above.
Here is an example of what your program should look like:

```
person: Ada
location: MVCDS
emotion: excited
noun: flashcards
verb: biking
noun: lunch
adjective: sleepy
verb: solve
adjective: smart
different person: Alan

Output:
Hello Ada,
  I have started learning Computer Science at MVCDS and I am really
  excited to get started. After working on my flashcards for this
  class, I will go biking with my friends after school. I just got
  back from lunch and I feel very sleepy. I hope to solve my homework
  soon, that would make me feel smart.
Talk to you later, Alan
```

Make sure to test your code!

Part 2: Snowfall at MVCDS

Its winter time at Northwestern Ohio, which means it's that time of year when every school has a snow day except MVCDS. Mr. Boehm said that he would close the school when the day's snowfall is twice as much as the total snowfall. Let's make a chart to keep track of our total snow.
Write a program that reads the number of inches of snow over the week at MVCDS. The days include Monday, Tuesday, Wednesday, Thursday, and Friday. We do not care about the weekend because we do not have school then. You must print the information using a table like the one found below. After that, compute the percent change in the amount of snow, in inches, on the ground between each day. Print the results of that as well.

Here is an example output of the program:

```
Please enter the snowfall per day below
Monday: 3
Tuesday: 1
Wednesday: 2
Thursday: 12
Friday: 2
```

Intro to CS in Python
Math Homework

```
Snowfall at MVCDS
- - - - - - - - - - - - - - -

Day   /Total Snow(in)  / Percent change %
Mon   /           3    /       0.0
Tues  /           4    /      33.3
Wed   /           6    /      50.0
Thur  /          18    /     200.0
Fri   /          20    /      11.1
```

Looks like on Thursday there was twice as much snow fell compared to the total snow (12 inches fell on Thursday while 6 inches had already fallen). MV had a snow day on Thursday. Wow!

How are we computing the percentages for each day? Let's do the change from Monday to Tuesday: `100*(total_snow - monday_snow)/(monday_snow)`. We put the numbers in and get `100*(4-3)/3`, which gives us `33.33333`... . This brings up the second problem: how do we stop it at the first decimal place? Here is some code to do this:

```
>>> print "The change is: %.1f percent" % (3.14159265)
The change  is: 3.1 percent.
```

Part3:

It's Friday so that means pizza for lunch. Chef Joe needs to order pizza from the Marco's and wants to get the most pizza possible for the students. Let's assume it costs $1.50 per slice and there are eight slices in a pizza. Chef Joe has a budget of $600 to use for Friday and wants to get the most pizza possible. Of course the best way to get the biggest pizza possible, but let's compare the different sizes and how many square inches of pizza we get when we order from Marco's.
We'll assume the diameter (the length from one side of the pizza to the other through the center) of the small, medium, large, and extra large pizzas are 8, 12, 16, and 18 inches respectively. With the amount of many he has and the price per pizza, how many square inches of pizza would he get if he got **all** smalls, **all** mediums, **all** larges, or **all** extra larges?

Here is the expected output:

```
Size of small: 8
Size of medium: 12
Size of large: 16
Size of extra large: 18
Money: 600

***************************
Size  /    area of a pie(in^2)   /    Total area (in^2)
S     /    50.26                 /    2513
M     /    113.09                /    1137
L     /    201.05                /    10052
XL    /    254.46                /    12723
```

Intro to CS in Python
Math Homework

As in the first two parts of this homework, you ask for input from the user for the sizes of the pizza and the amount of money they have. Knowing that a slice costs $1.5 and that there are eight slices in a pie, you can figure out the cost per pie (we are assuming that every pie costs the same amount). From there, you can see how many square inches of pizza you can buy depending if you get **all** of a certain size of pizza.

If you look closely, you can see that we converted the floats for the area of a pie to ints for the total area section. Remember to do that! Do not use "%.0f" to get your answer for this.

Bonus part:

This is a bonus in the sense of knowledge, not points. You will not get more points from doing this bonus, but you will work on your skills more to become a better programer.

Try to think of a few each features to add to your homework that you completed. Add a "largest amount of snowfall in one day" column to keep track of that record, or have a column that converts the square inches of pizza to square feet.