UNIVERSITY OF SASKATCHEWAN

Department of Computer Science
176 Thorvaldson Building
110 Science Place, Saskatoon, SK, S7N 5C9, Canada
Telephine: (306) 966-4886, Facimile: (306) 966-4884

Assignment 1

Programming Basics

Date Due: 5:00pm Friday Sept 27 **Total Marks: 20**

General Instructions

- **This assignment is individual work.** You may discuss questions and problems with anyone, but the work you hand in for this assignment must be your own work.
- Each question indicates what to hand in. You must give your document the name we prescribe for each question, usually in the form aNqM, meaning Assignment N, Question M.
- Make sure your name and student number appear at the top of every document you hand in. These conventions assist the markers in their work. Failure to follow these conventions will result in needless effort by the markers, and a deduction of grades for you.
- Do not submit folders, or zip files, even if you think it will help. It might help you, but it adds an extra step for the markers.
- Programs must be written in Python 3.5+
- You may not, in part or in whole, submit any code that was written by generative AI tools, such as ChatGPT
- **Assignments must be submitted to Canvas (the course website).**
- **Canvas will not let you submit work after the assignment hand-in closes.** It is advisable to hand in each answer that you are happy with as you go. You can always revise and resubmit as many times as you like before the deadline; only your most recent submission will be graded.
- Read the purpose of each question. Read the Evaluation section of each question.

Question 1 (7 points):

Purpose: To familiarize you with the Academic Honesty policy, and Canvas's assignment submission, and to do a little writing practice.

Read the Academic Honesty page at `https://www.cs.usask.ca/students/current-students/academic-honesty.php` (click coloured text to open). In the context of what you read, which of the following scenarios are violations of academic honesty? Explain in one or two sentences why or why not by referring to the academic honesty webpage. Please do not write lengthy answers.

(a) Red and Blue are working on an assignment together. Red emails her answer to Blue. Blue copies it and hands it in.

(b) Giovanni and Ash are enemies. Ash hacks into Giovanni's user account, and copies Giovanni's work for the assignment.

(c) Squirtle, Charry, Bulba, and Pikachu are a study group who have decided to work together on the assignment. Each of them answered one question, and then brought the answer to the group for discussion. After a long process of discussion and analysis, in which all the group members participated equally, they created a final answer to each question that they all copied.

(d) Brock and Misty just met in the lab. Misty has done some programming before, but Brock is a novice. Misty helped Brock fix his program so that it looked pretty much like her own.

(e) Jessie and James were passing through the lab when their friend Meowth stopped them and asked them for help figuring out why his program wasn't working right. James and Jessie looked at Meowth's program and explained what the problem was and what was causing it. Meowth thanked them (grouchily), and proceeded to fix the error on his own.

(f) Mewtwo and Lucario worked on the assignment together. Once they were fairly sure they understood the solutions, they destroyed all copies of their work. After playing an hour of Pokemon, they each went home and recreated the solutions to the assignment separately, from scratch, without consulting each other, or the notes they made.

(g) Magnemite was having trouble with one subtask in his program, and so he asked ChatGPT to provide some sample code for that task. After reading ChatGPT's answer carefully and trying to understand it, he copies the sample code back into his own program. Sure enough, it works, so he hands the program in.

Feel free to discuss these scenarios with other students, TAs, and instructors. It's very important that everyone understand the rules, as they will be applied to all assignments in this course. When the solutions are released, come back to these scenarios to make sure your understanding is consistent with ours!

What to Hand In

Hand in your answers in a file called `a1q1`. Allowed file formats are plain text (`.txt`), Rich Text (`.rtf`), and PDF (`.pdf`). We permit only these formats to ensure that our markers can open your files. Documents that cannot be opened conveniently will not be graded and receive 0 points.

Evaluation

- 1 mark for each correct answer. Answers should be justified by referring to the academic honesty policy. Writing style need not be formal, but should reflect an attempt at university-level English.

We are not grading for grammar or spelling. These are important, but there are other classes to evaluate that. However, there should be an attempt to make complete sentences, and students should avoid unprofessional abbreviations. Answers that cannot be understood because of poor writing will receive 0 marks.

Question 2 (3 points):

Purpose: To practice console input and output of strings.

A "mad-lib" is a fill-in-the blank game. One player writes a short story in which some words are replaced by blanks. For each word that is removed, the appropriate part of speech is noted: e.g. noun (person/-place/thing), adjective (word that describes a noun), verb (an action, e.g. eat), adverb (modifies a verb, e.g. quickly). Then, before reading the story, the story-writer asks the other player to write down a word of the appropriate part of speech for each blank without knowing the context in which it will be used. In this way, a humorous (sometimes) or non-sensical (usually) story is created.

Write your own mad-lib. It must have **at least three blanks** in it. Now write a Python program that does the following:

(a) Using the `input()` syntax (see the section "Reading Strings from the Keyboard" from the course readings) for reading strings from the console, prompt the user to enter a word of the appropriate part of speech for each blank in your program. Have a different, appropriately named variable refer to each word.

(b) Print your story to the console using the `print()` syntax, filling in the blanks in your story using the strings referred to by the variables that you gathered in Step 1. While it is possible to do this using a single `print()`, readability of code is important! You are permitted to use as many `print()` statements as you wish.

You must write your own unique story. There is no good reason why two students should submit the same story. Remember you must use a minimum of three blanks, and therefore, your program must read at least three words from the console. You can have more blanks, but get all your other work done before you spend a lot of time having fun with this question!

Sample Run

Here is an example of how your program's console output might look. **Do not submit this story!** Write your own story. In our example, we use green text to show what the user typed in; blue text highlights where the user's data gets put in the story. If you're using PyCharm to run your program, the user input will also be green, but it won't show any blue text.

```
Enter a verb ending in "ing":  zipping
Enter a noun:  squirtle
Enter a verb (past tense):  zapped
Pikachu was zipping through some tall grass.
Suddenly, a squirtle appeared!
The squirtle zapped Pikachu.
It was not very effective.
```

What to Hand In

A file called `a1q2.py` containing your finished program, as described above.

Evaluation

- 3 marks for reading an adequate number of words and correct behaviour of your story with those words

Question 3 (5 points):

Purpose: To practice the use of arithmetic expressions.

Phoenix Wright is a lawyer who runs a small law office with some number of assistants. When (if?) Phoenix gets paid for a case, he divides the money between himself and all his staff. The office contract states that Phoenix, as senior partner, takes 25% of the fee for himself and divides the rest equally among all his staff.

Write a program for Phoenix that will ask the user to input the payment for one case (in dollars), and the number of staff in the office (not including Phoenix himself). Then the program should display the value of Phoenix's share of the fee, the value of the remaining share to be divided amongst the staff, and the value of the pay that each staff member takes home.

You may assume that the user supplies valid input from the console, that is, a positive number for the fee, and a positive number for the staff size.

Sample Run

If completed correctly, your program's console output should look something like this. As usual, green text shows the text entered by the user, and blue text highlights the values calculated by the program.

```
Enter the payment for the case:  9000
Enter the number of staff:  2
Phoenix Wright's 25% share of the fee is worth: $2250.0
The staff's 75% share of the fee is worth: $6750.0
Each staff member takes home: $3375.0
```

Notice that the user is not expected to type the $ when entering the value of the legal fee. It is also not a concern if your displayed dollar amounts have more or less than two digits after the decimal place, or don't have a decimal at all. You will not be penalized for this (unless the values computed are incorrect). If you are having trouble getting the dollar amounts to print right next to the dollar signs without a space in between, think about how you might use the string concatenation operator (the very end of textbook Section 2.3.4 "Operators on Strings") to achieve the desired output.

What to Hand In

Hand in your solution in a file called `a1q3.py`.

Evaluation

- 4 marks for correct input reading and subsequent calculations
- 1 mark for somewhat-tricky formatting: making sure there is no space between dollar signs and the matching dollar amounts

Question 4 (3 points):

All programs that you submit for this assignment will be assessed with regard to their **style and readability**. Make sure that all of your code:

- includes your name, nsid, and student number as comments
- uses informative variable names
- makes effective use of formatting and white space
- includes a `docstring` for all **function definitions** in the program
- for longer programs, uses helpful and concise single-line comments where appropriate

What to Hand In

There is nothing to hand in for this question. It's simply describing how the rest of your work will be evaluated with regard to style and readability. It is possible to earn these marks even if you don't hand in every other question in this assignment, but you must hand in enough work such that your style and readability can be adequately assessed.

Evaluation

- 3 marks for consistently following the principles listed above

Question 5 (2 points):

In your submission for this assignment, you have an opportunity to earn something we'll call "the wow factor". To earn the wow factor, you must do something creative that goes beyond the expectations laid out in the assignment. This could be an extra feature for one of the previous questions, or it could be an extra little program that you submit that builds on the skills used in the assignment.

You should not aim to do a lot of work for this, but should instead demonstrate in a simple and effective way the depth of your understanding. In short, take ownership of your work and "impress us".

Telling you exactly what to do to earn "the wow factor" would defeat the entire purpose, but here are some very general ideas.

- improve the aesthetic appeal of your program(s)
- improve the user-friendliness of your program(s)
- remove a simplifying assumption and handle the resulting added complexity
- compare two different ways of doing the same task, and submitting data/analysis on which was better
- just doing something generally cool

One quick warning: using fancy extra code libraries in order to AVOID a key learning objective from the assignment is not impressive, it's actually anti-impressive. Focus instead on solving problems using simple clean foundational programming skills. Using extra libraries is fine if they let you do something BEYOND what the assignment asked, but not if they're replacing a core assigned task.

What to Hand In

A plain text file called `wow.txt` describing what you did that you think merits the wow factor. One or two sentences is enough. If you create any additional code files, hand those in too.

Evaluation

- 2 marks for something impressive