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 2

Functions

Date Due: 5:00pm Friday Oct 4 **Total Marks: 28**

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 (9 points):

Purpose: To distinguish between the different parts of functions and function calls

Below is a program that uses a function to compute the cost of a meal from Jimmy Boy's Catering Company which is priced in dollars. There is a fixed cost of $250 in supplies per meal course and a mandatory 15% gratuity automatically applied to the total.

```
1  def cost_of_meal(number_of_courses, number_of_diners, price_per_diner):
2      """
3      Computes the cost of a fancy meal in dollars
4
5      number_of_courses: the number of courses in the meal
6      number_of_diners: the number of diners attending
7      price_per_diner: cost for a single diner to attend in dollars
8
9      Returns: total cost of the meal
10     """
11     total_cost_of_supplies = number_of_courses * 250
12     total_diner_cost = number_of_diners * price_per_diner
13     subtotal = total_cost_of_supplies + total_diner_cost
14     gratuity = total_diner_cost * 0.15
15     return subtotal + gratuity
16
17 # 3 course meal for 100 pepole, at $42.50 each
18 first_meal = cost_of_meal(3, 100, 42.50)
19
20 # 5 course meal for 40 people, at $68.14 each
21 second_meal = cost_of_meal(5, 40, 68.14)
```

For each of the following terms, list **ALL** of them that you can find in this source code, along with their line numbers. Submitting a screenshot where you have clearly identified these elements is also acceptable.

(a) *Function definitions*

(b) *Function calls*

(c) *Function arguments*

(d) *Function parameters*

(e) What is the *scope* of:

  i. `gratuity`

  ii. `second_meal`

  iii. `price_per_diner`

(f) What are the values referred to by `first_meal` and `second_meal` when the program ends?

What to Hand In

Hand in your answers in a file called `a2q1`. Allowed file formats are plain text (`.txt`), Rich Text (`.rtf`), and PDF (`.pdf`).

Evaluation

- 1 mark for each of parts (a) through (d)
- 3 marks for part (e), one for each scope correctly identified
- 2 marks for part (f), one for each correct final value

Question 2 (3 points):

Purpose: To write a function that performs a subtask and returns an answer.

When creating function definitions, it is important for a function to receive its **input data** via its **parameters**. It is almost **always a bad idea** for a function that performs calculations to also perform any kind of console input or output. In this question, you'll apply that principle to a simple problem.

Fuel Cost Function

For this question, write a function to help you calculate the fuel cost you'll need to pay for a roadtrip.

Write a Python function that takes three parameters as input and **RETURNS** the **total fuel cost of the drive**. The three parameters should be the **distance of the trip** in km, the **price of fuel** in dollars per liter, and the **fuel efficiency** of the vehicle. Unfortunately, the vehicle is of American make, and so the fuel efficiency is measured in **miles per gallon (MPG)**. Your function will need to do the relevant unit conversions to make sure its calculation is correct.

Importantly, your function should do no console input and no console output. It must receive its input through its parameters and send its output using a return value.

Console I/O

In the main part of your program (i.e. NOT inside the function's body), use console input to get the necessary values needed by your function. You may assume that the user supplies valid input. Then call your function using those values to get the total cost of the trip. Finally, display a message that indicates the total cost.

Sample Run

Here is an example of how your program's console output might look. Green text was entered by the user; blue text came from data returned by the function.

```
Roadtrip Calculator
Enter distance of trip (in km): 480
Enter cost of fuel (per liter): 1.5
Enter efficiency of vehicle (in MPG): 35.9
The total fuel cost of your roadtrip is $47.17408203196272
```

What to Hand In

Hand in your solution in a file called `a2q2.py`.

Evaluation

- 1 mark for correct output
- 2 marks for correct usage of a function

Question 3 (5 points):

Purpose: To write a function that uses helper functions inside another function

A **helper function** is a function that is meant to be called from some other "bigger" function. It's a technique for dividing a problem into parts, and thus very important in computer science. For this question, you'll create a function that uses two other helper functions to do its work.

Painting a House

Mrs. Peacock loves the color blue. She wants to paint the entire outside of her house blue, including the roof!

The house has four walls, and all of the walls are the same size. With W as the width of each wall and H as the height of each wall, the formula for the total outer wall area is:

$$walls = W * H * 4$$

The roof is a **pyramid**, which is to say that it is made up of four isosceles triangles. The width of the **base** of this pyramid must be W, i.e. the width of each wall. If we let T be the height of each of the triangles that make up the pyramid (in geometry this is often called the *slant height*), then the formula for the total roof area is:

$$roof = 4 * W * T/2$$

All of the above units of size are in meters. If we are also told the cost of paint per square meter, then our task is to calculate the total cost for the paint needed to paint the house.

Helper Functions

First, write the following **two helper functions**. They will be very short and simple. That's ok; in fact, that's the point.

- A function to calculate the area of the walls
- A function to calculate the area of the roof

In both cases, the functions should take only the parameters they need to perform their computations.

Total Cost Function

Next, write a function to compute and **return** the total cost of paint needed to paint the house. The function should have four parameters:

- the width of the walls
- the height of the walls
- the slant height of the roof (i.e. the height of each triangle)
- the cost per sq. m. of paint

This function should **call the helper functions** to perform their part of the calculation.

Console I/O

In the main part of your program (i.e. NOT inside the function's body), use console input to get the values needed by your function. You may assume that the user supplies valid input. Then call your function using those values to get the total cost of painting the house. Finally, display a message that indicates the total cost.

Sample Run

Here is an example of how your program's console output might look. Green text was entered by the user; blue text came from data returned by the function.

```
House Paint Calculator
Width of house walls: 4
Height of house walls: 5
Roof slant height: 3
Cost of paint for each square meter: 2
Total cost for an all-blue house: $208.0
```

What to Hand In

Hand in your solution in a file called `a2q3.py`.

Evaluation

- 2 marks for correct output
- 3 marks for correct function design and use of helper functions

Question 4 (6 points):

Purpose: To write functions meant to be used in sequence

A common problem-solving pattern is to call several functions in sequence, with the **return value(s)** of each function call used as the **argument(s)** for the NEXT function call. You'll practice this technique here.

Registering an online account

Our overall goal is to allow a user to register an account name for an online service. But we'll divide the work up into 3 separate functions, which the main part of your script will then call in order.

Normally, when you are breaking up a problem into parts, it's up to you to decide how many functions to have and what their inputs and outputs should be. However, since this is for practice, in this case just be sure to read the instructions carefully and follow them exactly.

Function 1: Getting the user's account name and date of birth

Write a function that uses the `input()` function to obtain the user's desired account name and their date of birth (in MM/DD/YYYY format). You may assume the user enters the date in the correct format. Note that it's ok to put console input inside a function like this, so long as it's the MAIN purpose of the function!

The function should then return both of these values. In Python, having multiple return values is easy! Just separate the values you want to return with commas after the return statement, and when calling the function, put multiple variables on the left hand of the equal sign to "collect" each return value. Here is an example:

```
def coords():
    return 10, 20

x, y = coords()
```

Function 2: Formatting the account name

Many online services have formatting rules regarding acceptable usernames. For your second task, write a function that accepts the original username and returns a formatted version of that name according to the following rules:

- The name should be truncated to be at most 10 characters long. Any characters beyond the 10th are lost
- All letters should be converted to lower case
- All space characters (`" "`) should be converted to underscores (`"_"`)

You may find the Python string methods `.lower()` and `.replace()` to be useful here.

Function 3: Finalizing the account name

For most online services, it is very common for multiple people to request the same name. To get around this, users often add their year of birth to the end of their username.

Your final function should accept the now-formatted account name and the user's birthdate, and append the **last two digits of the user's birth-year** to the account name. Recall that the birthdate was entered in MM/DD/YYYY format. The function should then return the now-modified account name.

Main program

The "main" part of your program should call each function in sequence, passing along their return values as needed, and then display a message to the user indicating the finalized account name.

Sample Run

Here is an example of how your program's console output might look. Green text was entered by the user.

```
Enter your desired account name: Pikachu Guru
Enter your date of birth (MM/DD/YYYY): 5/22/1986

Your account has been created.
Your account handle is: pikachu_gu86
```

What to Hand In

Hand in your solution in a file called `a2q4.py`.

Evaluation

- 2 marks for correct output
- 4 marks for correct function design and use of function sequence

Question 5 (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 6 (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