UNIVERSITY OF
SASKATCHEWAN

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

CMPT 141
Winter 2024
Introduction to Computer Science

# Assignment 2

## Functions

**Date Due: 11:59pm Friday Jan 26** **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+.
- **Assignments must be submitted to Canvas.** There is a link on the course webpage that shows you how to do this.
- **Canvas will not let you submit work after the assignment deadline.** 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 leaving dogs at Joey's Doggy Day Care. There is a cost per dog that is charged daily.

```
 1  def day_care_cost(number_of_dogs, number_of_days, price_per_dog):
 2      """
 3      Computes the total cost of a stay at Doggy Day Care
 4
 5      number_of_dogs: the number of dogs
 6      number_of_days: duration of stay in days
 7      price_per_dog: cost for a single dog in dollars
 8
 9      Returns: total cost of the stay
10      """
11      daily_dog_cost = number_of_dogs * price_per_dog
12      total_cost = daily_dog_cost * number_of_days
13      return total_cost
14
15  # compute the cost of a stay for 3 dogs
16  # staying 7 days at a rate of $49.50 per dog
17  first_stay = day_care_cost(3, 7, 49.50)
18
19  # compute the cost of a stay for 10 dogs
20  # staying 3 days at a rate of $19.75 per dog
21  second_stay = day_care_cost(10, 3, 19.75)
```

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. `daily_dog_cost`

  ii. `number_of_days`

  iii. `first_stay`

(f) What are the values referred to by `first_stay` and `second_stay` 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`), or PDF (`.pdf`). PNG and JPG are also acceptable if you submit a screenshot or image with drawings on it.

### 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 variable value correctly identified
- -1 mark if identifying information is missing (name, NSID, student number and instructor's name)

## Question 2 (4 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.

# Trip Cost Function

For this question, write a function to help a travel agent calculate the cost of a trip for a family. The trip includes a flight and some number of nights in a hotel. The hotel sleeps at most 2 people in each room.

Write a Python function that takes four parameters as input and **RETURNS** the **total cost of the trip**. The four parameters should be the **airfare per person** in dollars, the **room cost per night** in dollars, the **number of travellers**, and the **number of nights**.

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.

```
Calculate Trip Cost
Enter cost of flight per person ($): 1000
Enter cost of a double room per night:100.00
Enter the number of people: 11
Enter the number of nights:7
The total cost of the trip for the group is  $15200.0
```

## 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
- 1 mark for code readability and style

## Question 3 (7 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
- 2 marks for code readability and style

## Question 4 (8 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.

### Generating a Username

Our overall goal is to generate a partially randomized **username** based on a user's personal initials. 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 name

Write a function that uses the `input()` function to obtain the user's first, middle, and last name via console input. The user can leave some names blank (which means the matching name will get set to the empty string). 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 program should format each name such that (no matter what the user types) the first letter of each name is capitalized and the rest are all lower case (hint: use the string method `.title()`).

Finally, the function should return all 3 names. 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:

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

x, y = coords()
```

### Function 2: Converting to initials

Your next function should accept the first, middle, and last name as parameters, and return that person's initials (i.e. the first letter of each of their names) as a single string.

Recall that one (or more) of the names may have been left blank. If this happens, then use a capital letter "X" for that name in the person's initials. For example, `"Ash Ketchum"` should have initials of `"AXK"`, and `"Pikachu"` should have initials of `"PXX"`.

### Function 3: Generating the username

Your final function should accept a person's initials as a parameter and return a username consisting of the initials converted to lower case and followed by a random 3 digit number between 100 and 999 (inclusive). Hint: import the `random` module and use the `.randint()` method.

### 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 their generated username.

## 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.

```
Please enter your name at the prompts.
You can leave them blank if needed (e.g. if you don't have a middle name)
Your first name: ash
Your middle name:
Your last name: ketchum

Your setup is complete, Ash  Ketchum
Your login ID is: axk763
```

## 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
- 2 marks for code readability and style