Assignment 1

Course: CS65 - Introduction to Computer Science (Spring 2022)
Instructor: Md Alimoor Reza, Assistant Professor of Computer Science, Drake University
Due: Thursday March 03, 11:50 PM

Introduction

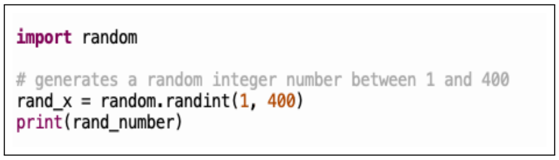
Selection statements (if/elif/else combinations) allow us to execute different code based on the current values of variables defined in your program. This assignment will give you practice with this specific feature. Your programs should be written with multiple user-defined functions in the same file. This helps break down the task into smaller and simpler pieces. Functions also make it easier to test individual components, as parts of the program can be tested separately.

Logistics

In this assignment, You will be creating a python file for each task separately (with a **.py* extension), where you will save your python instructions. By default, Thonny opens an unnamed file in the a *text editor* (top text pane). It is an excellent practice to write a formal header and suppress it as comments. A line of code with a preceding # is considered as a comment in Python. You should type in the necessary parts as shown below, e.g., *your name, your contact email, description, etc.*. Save the file using the format as follows *firstname_lastname_a1_task*.py* (all in lowercase letters). For example, I saved my file as *md_reza_a1_task1.py*. I have assigned you to a group. You should only work together but can submit a separate copy of the assignment.

Generating Random Numbers

Recall from your lecture notes and class discussion; there is a library in Python to generate random numbers Like math module or graphics module, you can import random module to get access to random number generating functions. To generate a random number, you have to import the random module. Then you can generate a random integer number between a range of values denoted by a *lower_range* and an *upper_range*. For example, in order to generate a random integer number between *lower_range* of 1 and *upper_range* of 400, we need to do the following:



```
import random

# generates a random integer number between 1 and 400
rand_x = random.randint(1, 400)
print(rand_number)
```

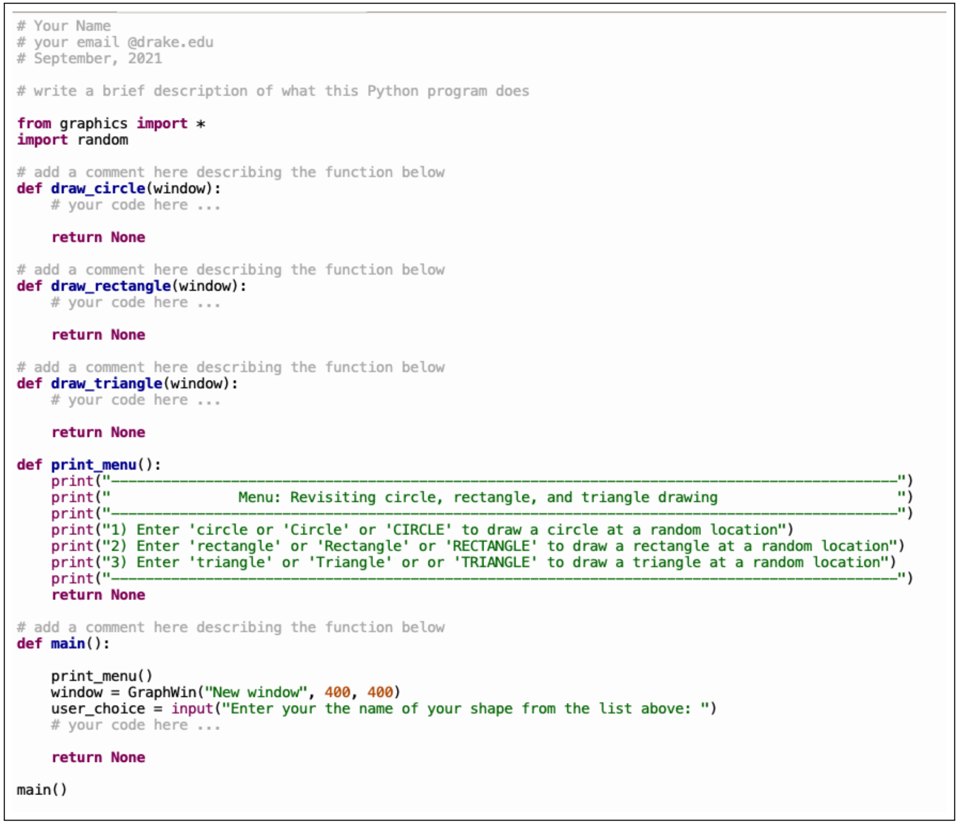
You may find this Python code snippet useful for solving the tasks below.

Task 1 (60 points): Revisiting circle, rectangle, and triangle drawing

In your lab3, you have written a python programs that can draw **shapes** of different sizes and at different locations, where a shape could be either i) **circle**, ii) **rectangle**, or iii) **triangle**. For each shape, you were asked to write a separate user-defined function. More specifically, you have a function for drawing **circles**, another function for drawing **rectangles** and finally, one function for drawing **triangles**. Write a program that can draw any of the above three shapes based on user input. More specifically, your program should take user input. User should type one of the followings: i) **circle**, ii) **rectangle**, or iii) **triangle**. Based on the user input, your program should do the following:

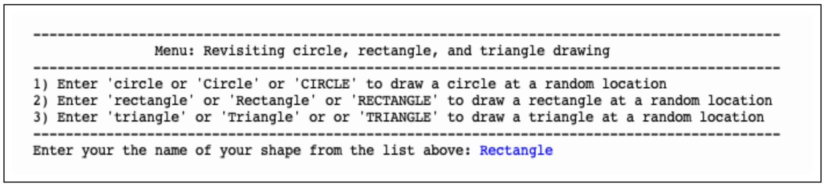
- **Circle**: if the user enters *circle* or *Circle* or *CIRCLE*, your program should draw a circle at a *random location* with a fixed radius of 100 pixels.
- **rectangle**: if the user enters *rectangle* or *Rectangle* or *RECTANGLE*, your program should draw a rectangle at a *random location.*
- **triangle**: if the user enters *triangle* or *Triangle* or *TRIANGLE*, your program should draw a *triangle* at a *random location.*

You should also fill the background color of your shape with any color of your choice, e.g., *red, green,* or blue. You could start with the provided code below, which has an empty but reasonably good structure. Note that you could download this empty structure from Blackboard.

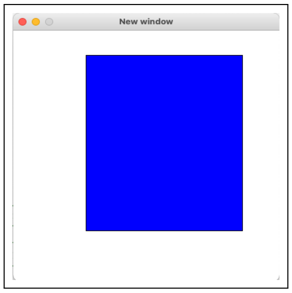


```python
# Your Name
# your email @drake.edu
# September, 2021

# write a brief description of what this Python program does

from graphics import *
import random

# add a comment here describing the function below
def draw_circle(window):
    # your code here ...

    return None

# add a comment here describing the function below
def draw_rectangle(window):
    # your code here ...

    return None

# add a comment here describing the function below
def draw_triangle(window):
    # your code here ...

    return None

def print_menu():
    print("------------------------------------------------------------------------------------------")
    print("                  Menu: Revisiting circle, rectangle, and triangle drawing                ")
    print("------------------------------------------------------------------------------------------")
    print("1) Enter 'circle or 'Circle' or 'CIRCLE' to draw a circle at a random location")
    print("2) Enter 'rectangle' or 'Rectangle' or 'RECTANGLE' to draw a rectangle at a random location")
    print("3) Enter 'triangle' or 'Triangle' or or 'TRIANGLE' to draw a triangle at a random location")
    print("------------------------------------------------------------------------------------------")
    return None

# add a comment here describing the function below
def main():

    print_menu()
    window = GraphWin("New window", 400, 400)
    user_choice = input("Enter your the name of your shape from the list above: ")
    # your code here ...

    return None

main()
```

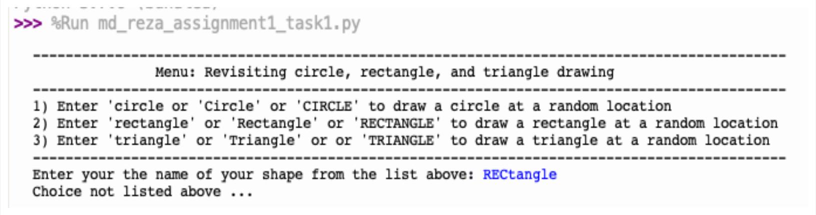
When you enter any valid user input, your program should show the shape of your choice. For example, for the choice below, it should draw the rectangle at a random location like the one below:



```
------------------------------------------------------------------------------------------
                  Menu: Revisiting circle, rectangle, and triangle drawing
------------------------------------------------------------------------------------------
1) Enter 'circle or 'Circle' or 'CIRCLE' to draw a circle at a random location
2) Enter 'rectangle' or 'Rectangle' or 'RECTANGLE' to draw a rectangle at a random location
3) Enter 'triangle' or 'Triangle' or or 'TRIANGLE' to draw a triangle at a random location
------------------------------------------------------------------------------------------
Enter your the name of your shape from the list above: Rectangle
```



Your program should print a message when the user enters an invalid choice of shape, e.g., see the example below:

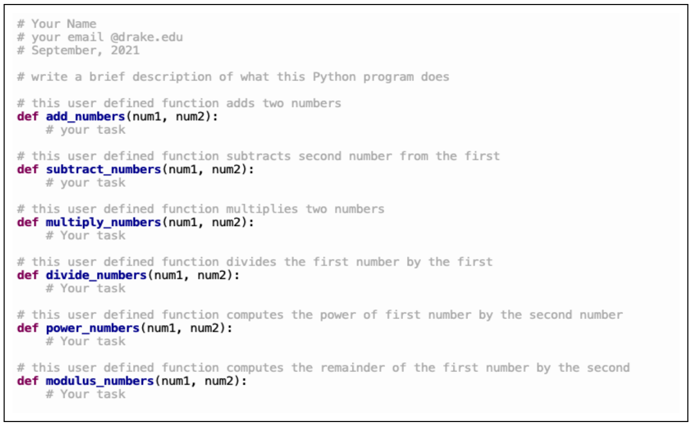


```
>>> %Run md_reza_assignment1_task1.py

  --------------------------------------------------------------------------------------
                 Menu: Revisiting circle, rectangle, and triangle drawing
  --------------------------------------------------------------------------------------
  1) Enter 'circle or 'Circle' or 'CIRCLE' to draw a circle at a random location
  2) Enter 'rectangle' or 'Rectangle' or 'RECTANGLE' to draw a rectangle at a random location
  3) Enter 'triangle' or 'Triangle' or or 'TRIANGLE' to draw a triangle at a random location
  --------------------------------------------------------------------------------------
  Enter your the name of your shape from the list above: RECtangle
  Choice not listed above ...
```

Task 2 (40 points): Revisiting simple arithmetic operations

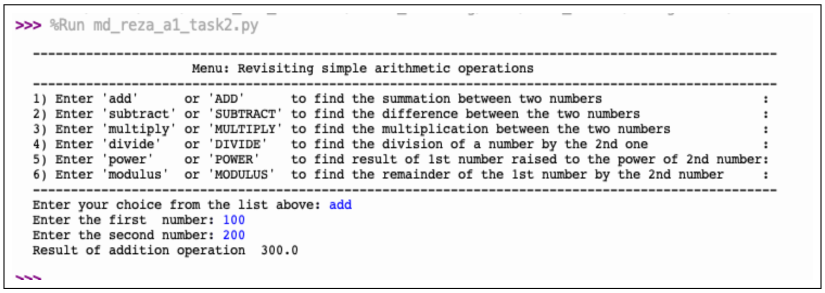
In this task, you will be revisiting one of the problems you solved in "Lab 2". You could reuse the functions you wrote for that task. In addition to writing functions that calculate simple arithmetic operations such as *add, subtract, multiply,* and *divide* between two numbers, you will need to add two more functions *power* and modulus. Your program should display a question that determines what operation the user is interested in. The choices are as follows:

- **add**: if the user enters *add* or *ADD*, your program should prompt the user to enter two more numbers, and then *call* the appropriate function to compute the result.
- **subtract**: if the user enters *subtract* or *SUBTRACT* your program should prompt the user to enter two more numbers, and then *call* the appropriate function to compute the result.
- **multiply**: if the user enters *multiply* or *MULTIPLY*, your program should prompt the user to enter two more numbers, and then *call* the appropriate function to compute the result.
- **divide**: if the user enters *divide* or *DIVIDE* or , your program should prompt the user to enter two more numbers, and then *call* the appropriate function to compute the result. Extra credit: If the second number is zero, your program should not compute the result instead it should print an output message "ERROR: divide by zero."
- **power**: if the user enters *power* or *POWER*, your program should prompt the user to enter two more numbers, and then *call* the appropriate function to compute the result. For example, if the user enters 2 followed by 3, then the expected result is 8. If the user enters 3 followed by 2, then the expected result is 9, and so on.
- **modulus**: if the user enters *modulus* or *MODULUS*, your program should prompt the user to enter two more numbers, and then *call* the appropriate function to compute the result. This function should compute the remainder of two numbers. For example, if the user enters 15 followed by 4, then the expected result is 3. If the user enters 15 followed by 3, then the expected result is 0, and so on. Extra credit: If the user enters any negative number, you should convert it back to a positive one before computing the remainder. You will receive extra points for this conversion. You will also receive extra points if you can compute the remainder by not using the % operator.



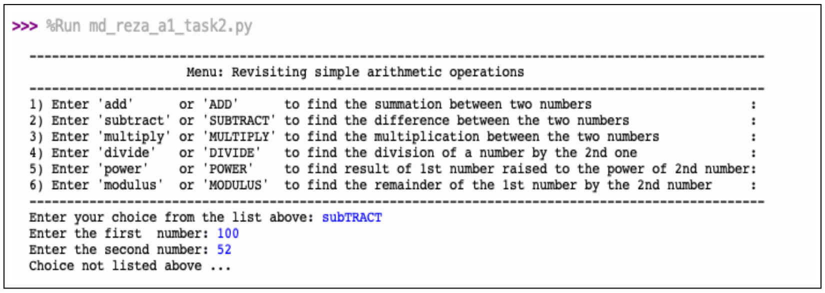
```
# Your Name
# your email @drake.edu
# September, 2021

# write a brief description of what this Python program does

# this user defined function adds two numbers
def add_numbers(num1, num2):
    # your task

# this user defined function subtracts second number from the first
def subtract_numbers(num1, num2):
    # your task

# this user defined function multiplies two numbers
def multiply_numbers(num1, num2):
    # Your task

# this user defined function divides the first number by the first
def divide_numbers(num1, num2):
    # Your task

# this user defined function computes the power of first number by the second number
def power_numbers(num1, num2):
    # Your task

# this user defined function computes the remainder of the first number by the second
def modulus_numbers(num1, num2):
    # Your task
```

When you enter any valid user input, your program should proceed to finish the calculation of your choice. More precisely, your program should first display a menu of available choices and then get a responses from the user as follows:



```
>>> %Run md_reza_a1_task2.py

  --------------------------------------------------------------------------------------------------
                      Menu: Revisiting simple arithmetic operations
  --------------------------------------------------------------------------------------------------
  1) Enter 'add'      or 'ADD'      to find the summation between two numbers                     :
  2) Enter 'subtract' or 'SUBTRACT' to find the difference between the two numbers                :
  3) Enter 'multiply' or 'MULTIPLY' to find the multiplication between the two numbers            :
  4) Enter 'divide'   or 'DIVIDE'   to find the division of a number by the 2nd one               :
  5) Enter 'power'    or 'POWER'    to find result of 1st number raised to the power of 2nd number:
  6) Enter 'modulus'  or 'MODULUS'  to find the remainder of the 1st number by the 2nd number     :
  --------------------------------------------------------------------------------------------------
  Enter your choice from the list above: add
  Enter the first  number: 100
  Enter the second number: 200
  Result of addition operation  300.0

>>>
```

You should take inspiration from the task 1 to display a menu of available choices. You should do all your computations as floating point operations instead of integer calculation. Like task 1, your program should print a message when user enters an invalid choice of operation (see the example below):



```
>>> %Run md_reza_a1_task2.py

  --------------------------------------------------------------------------------------------------
                      Menu: Revisiting simple arithmetic operations
  --------------------------------------------------------------------------------------------------
  1) Enter 'add'      or 'ADD'      to find the summation between two numbers                     :
  2) Enter 'subtract' or 'SUBTRACT' to find the difference between the two numbers                :
  3) Enter 'multiply' or 'MULTIPLY' to find the multiplication between the two numbers            :
  4) Enter 'divide'   or 'DIVIDE'   to find the division of a number by the 2nd one               :
  5) Enter 'power'    or 'POWER'    to find result of 1st number raised to the power of 2nd number:
  6) Enter 'modulus'  or 'MODULUS'  to find the remainder of the 1st number by the 2nd number     :
  --------------------------------------------------------------------------------------------------
  Enter your choice from the list above: subTRACT
  Enter the first  number: 100
  Enter the second number: 52
  Choice not listed above ...
```

What to turn in

You should submit: Your source code (two python files).

Grading Rubric:

Submitted correctly	05
Code is well commented	05
Boolean/branch/function usage	45
Code produces correct outputs	45
Total	100

In addition, there is 10 more points for the extra credit questions.