ICS3U Case Study: Surface Area Calculator

For this case study, we will look at a program that can be used to determine the total surface area of a composite shape – one build up of three-dimensional objects, such as cubes and cylinders. We will need to work out the areas of basic shapes, such as rectangles and circles, to do this. Code is below. Try it.

```
import math

def area_circle(radius):
    area = math.pi * radius**2
    return area
def area_rectangle(length, width):
    area = length * width
    return area
def area_triangle(base, height):
    area = base * height / 2
    return area
def get_integer(low=-math.inf, high=math.inf, prompt="Enter a value: "):
    integer = int(input(prompt))
    while integer < low or integer > high:
        integer = int(input("Invalid value, try again: "))
    return integer
def menu():
    print("What would you like to do: ")
    print("1. Add circle.")
    print("2. Add rectangle.")
    print("3. Add triangle.")
    print("4. Display total surface area.")
    print("0. Quit.")
    action = get_integer(low=0, high=4, prompt="Select an action: ")
    return action

# MAIN PROGRAM ===============================================================
print("Welcome to the surface area calculator.")
action = menu()
surface_area = 0
while action != 0:
    if action == 1:
        radius = get_integer(low=0, prompt="Enter the radius: ")
        surface_area += area_circle(radius)
    elif action == 2:
        length = get_integer(low=0, prompt="Enter the length: ")
        width = get_integer(low=0, prompt="Enter the width: ")
        surface_area += area_rectangle(length, width)
    elif action == 3:
        base = get_integer(low=0, prompt="Enter the base: ")
        height = get_integer(low=0, prompt="Enter the height: ")
        surface_area += area_triangle(base, height)
    elif action == 4:
        print("The surface area is", round(surface_area, 2), "units^2.")
    action = menu()
print("Goodbye.")
```

Let's start by examining the functions that are used in the program. There are three simple shapes that we use to calculate surface area: circles, rectangles and triangles. Each has its own function where the surface area for that shape is calculated. It is possible to write these functions in other ways. For example, since the area of a triangle is exactly half of a rectangle with the same overall dimensions, we could have done the following.

```
def area_triangle(base, height):
    area = area_rectangle(base, height) / 2
    return area
```

This isn't any easier to understand, or to calculate, and so we have used the familiar mathematical formula A=bh/2 instead.

The get_integer function is one that we have seen before, and tested. This leaved one last function, menu, whose only purpose is to display the available options to the user and to obtain their choice. It takes no arguments, as no values are required for it to work properly. We could have had the user select their choice in the main program, rather than in the function itself, but the decision was made to include it in the function to make maintenance easier – any changes to the menu system would likely require updating the arguments to get_integer, so keeping the code together makes sense.

The main program delegates all calculations to the various functions. Its purpose is to repeatedly show the menu (using a while loop) and update the running total for the surface area. This is generally a good way to structure non-trivial programs: keep the main program general, while assigning all major tasks and calculations to specific functions. Not only does this encourage the programmer to keep all functions focused on one or two simple tasks, but it also makes the main program much more readable.

As long as the user does not choose to quit the program, the condition action != 0 is True and the loop will execute. Inside of the loop, the program checks which option the user selected and executes the appropriate function. Any value that is returned is added to the running total, surface_area. If the user opts to display the value, it is rounded to two decimal places for convenience. Once the surface area has been updated, or displayed, the user is prompted to enter a new action via the menu.

Implement the following changes in your program.

- Add additional shapes, such as trapezoids and parallelograms. For each new shape, add a new function that will calculate its area. Don't forget to update the menu.
- Add the ability to remove shapes. For example, the surface area of a square with a circular hole cut into it may be calculated by finding the area of the square then subtracting the area of the circle.
- Using this program as a guide, write a new program that will calculate the total volume of a composite object (made up of prisms, cylinders, cones, etc.).