Computer Science 220L
Laboratory 8 – Conditionals

Learning objectives:

- Use conditional (if) control structures.
- Continue to practice graphics programming.
- Write functions

Getting started.
Organize yourselves into teams of two. Download the program rectangle.py.

1. Rectangles
In an earlier lab, you wrote a program called rectangle.py to draw a rectangle from its corners and to compute its area and perimeter (Ex. 9, Chapter 4). A copy with the code broken into functions is provided. The function getArea(r) has not been written. Write the function getArea() that accepts a rectangle, r, as a parameter and returns the area of the rectangle. Note getWidth() and getHeight() have been written. Add code to main() to test getArea().

Modify the program to draw three rectangles. Make them any color you want except red. Stop the program at this point and ask the user to click to continue.

Compute the areas of all three rectangles, but don’t display the areas. Determine which of the rectangles has the largest area, turn that rectangle red, and display

"The largest rectangle is number N."

Ask the user to click to end the program, and close the window at the end.

Swap driver and navigator.

2. Leap Year
There are approximately 365¼ days in a year. So that the calendar stays accurate, an extra day is added every four years.

A year is a leap year if it is evenly divisible by 4. However, that adds too many days, so a year is not a leap year if it is evenly divisible by 100. But then the year is a bit too short, so a year that is evenly divisible by 400 is a leap year. (So February 29, 2000, was a very special day. There won't be another leap day in a century year until 2400.)

Write leapYear.py. This program does not need to be graphics based. It should include a function isLeapYear(y) that accepts a year, y, and returns True if the year represents a leap year and False if it does not.

Swap driver and navigator.
3 A Boolean Function – Valid Telephone Number
Write a Python program, validPhone.py. main() prompts the user for a telephone number. main() calls a function, valid(n), that determines whether the number is in the correct format: ddd-ddd-dddd, where each d is a digit, and returns True or False. main() displays whether the number is valid.

Once the program works for one number, add a loop that asks for more numbers until the user presses enter without having entered data.

If you are pair programming, swap driver and navigator.

4. Triangles with Functions
In an earlier lab, we computed the area of a triangle from the lengths of its sides. We ignored the fact that not all side lengths can make a triangle. Now we'll write a better triangle program by writing several functions for the program triangle.py. main() begins by prompting the user for three side lengths, called a, b, and c.

Write main() in stages, testing each function before going on to the next. First test whether the three sides can form a triangle and report the result. Then, if the sides can form a triangle, compute the area. Then determine the shape. Then determine whether it is a right triangle.

Notes: (1) You don't need a function to determine scalene (why not?). (2) An equilateral triangle is also isosceles, but display only one of the two shapes. (3) Either an isosceles or a scalene triangle can be a right triangle, so this is a separate question from the shape question.
Write the following functions.

- **isATriangle(a, b, c) → boolean**: This function accepts the lengths of three sides and determines whether the three sides can form a triangle. Three sides can make a triangle if and only if all three are positive and the sum of any two sides is greater than the third. The function returns True if the sides form a triangle or False if they do not form a triangle.

- **area(a, b, c) → float**: This function accepts the lengths of three sides and returns the area of the triangle as a float. Compute the area using these formulas:

  \[ s = \frac{a + b + c}{2} \]

  where \( a, b, \) and \( c \) are the lengths of the sides, and

  \[ Area = \sqrt{s(s-a)(s-b)(s-c)} \]

- **isEquilateral(a, b, c) → boolean**: This function accepts the lengths of three sides and determines whether the triangle is equilateral. A triangle is equilateral if all three sides are equal. Return a boolean.

- **isIsosceles(a, b, c) → boolean**: This function accepts the lengths of three sides and determine whether the triangle is isosceles. A triangle is isosceles if any two sides are equal (note: equilateral triangles are also isosceles). Return a boolean.

- **isRight(a, b, c) → boolean**: This function accepts the lengths of three sides and determines whether the triangle is a right triangle. A right triangle means that the triangle contains a 90-degree angle. Right triangles satisfy the Pythagorean Theorem: \( a^2 + b^2 = c^2 \) where \( a, b, \) and \( c \) are the lengths of the sides. Remember that you don’t know what order the sides will be entered, i.e., you don’t know which sides are \( a, b \) and \( c \). Return a boolean.

  Sample test sets you might try: 1, 1, 1; 0, 2, 2; 3, 4, 5; -3, -4, 5; 12, 1, 14; 5, 5, 6; 8, 8, 8; 6, 5, 5.

Upload the files (don't forget to submit):
  Upload to the accounts of both students.

```python
rectangle.py          leapYear.py
validPhone.py         triangle.py
```