

University of Western Ontario, Computer Science Department
CS1026A SU19, Computer Organization

Assignment 1

Due: May 17, 2019

General Instructions: This assignment consists of 4 pages, 1 exercise, and is marked out of 100. Assignments are the independent work of each student. Software may be used to detect cheating.

Non-Functional Code Instructions:

1. Include brief comments in your code. Identifying yourself (the code's author) by name and user ID in the initial comment header. Comment also on key instructions and calculations in your code.
e.g.

```
##  
# A program for computing areas with inscribed circles.  
# Student Name: Alex Brandt  
# Student ID: abrandt5
```
2. Follow *good coding style* and normal Python conventions. This includes, but is not limited to:
 - (i) Meaningful variable names.
 - (ii) Conventions for naming variables and constants.
 - (iii) Use of constants over “magic numbers”.
 - (iv) *Readability*: indentation, appropriate white space (blank spaces) within instructions, consistency in the use of all of the above.

Evaluation:

1. Functional Requirements:
 - (i) Does your program correctly implement an algorithm for solving the problem?
 - (ii) Does your program behave according to the specifications?
 - (iii) Does your program handle invalid input?
 - (iv) Does your program output values according to the specifications?
2. Non-Functional Requirements (above).
3. Ability to follow directions precisely.

Submission Instructions: Your submission should include exactly two files (zipped into a zip file, if you'd like). These two files are: `userid_Assign1.py` and `userid_Assign1.pdf`; where `userid` is replaced by your UWO User ID (everything preceding “@” in your UWO email; e.g. `abrandt5`). Your python file contains the code which implements the algorithm for solving the following problem. Your PDF file contains the written portion of the following problem. Use proper English, spelling, and grammar. If the person correcting your assignment is unable to easily read or interpret your answer then it may be marked as incorrect without the possibility of remarking.

Problem 1. This problem deals with *inscribed circles*. An inscribed circle is a circle which fits maximally (with respect to area) inside another shape. In this problem we are interested in circles inscribed within squares and triangles. Figure 1 shows a circle inscribed inside of a square. Figure 2 shows a circle inscribed inside of a triangle. In this problem you must devise an algorithm (try not to google it!) for determining the area of the square (resp. triangle) excluding the area of the inscribed circle. That is, the coloured regions in each figure (the blue region in Figure 1 and the green region in Figure 2). Let us call the algorithm which computes areas with respect to the square the “square algorithm”. Let us call the algorithm which computes areas with respect to the triangle the “triangle algorithm”.

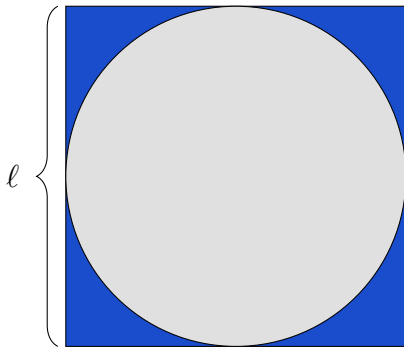


Figure 1: A circle inscribed inside of a square.

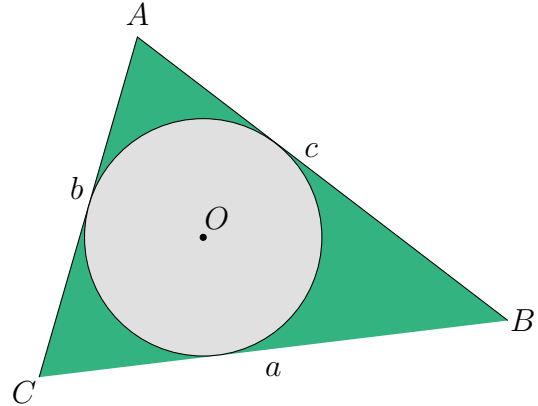


Figure 2: A circle inscribed inside of a triangle.

The written portion of this assignment is to define **pseudo-code** for each of those algorithms.

1. The square algorithm takes as input the side length ℓ of the square.
2. The triangle algorithm takes as input the 3 pairs of numbers $(x_A, y_A), (x_B, y_B), (x_C, y_C)$ which define the Cartesian coordinates of the vertices of the triangle.
3. **Your algorithm and subsequent program only needs to handle non-negative integer inputs.**
4. In your pseudo-code work generically with variables and not *instances* of the problem.

The programming portion of this assignment is to implement both algorithms within the **same** program. To facilitate this consider the following functional specifications for your program.

1. The program should begin by prompting the user with the question “Is your circle inscribed in a square or a triangle?” and receiving some string as input.
2. If the user responds with “s”, “S”, “square”, “Square”, or any combination of uppercase and lowercase letters which spell the word “square”, then your program computes the “square algorithm”.

3. If the user responds with “t”, “T”, “triangle”, “Triangle”, or any combination of uppercase and lowercase letters which spell the word “triangle”, then your program computes the “triangle algorithm”.
4. If the user’s input does not match any of the previous two cases then your program should print “Sorry, your input was: <input>. This program accepts one of "square" or "triangle"” where <input> is replaced by the actual input of the user. The program should then terminate without computing anything.
5. For the square algorithm:
 - (a) The program should then prompt the user with “The square’s side length: ” and then receive as input a positive integer which is the square’s side length.
 - (b) Compute and print the area of the inscribed circle to a precision of 3 digits after the decimal place. Include a meaningful message describing what the value is (e.g. **The area of the inscribed circle is: 23.345**).
 - (c) Compute and print the area of the square excluding the area of the inscribed circle (i.e. the blue shaded region) to a precision of 3 digits after the decimal place. Include a meaningful message describing what the value is.
6. For the triangle algorithm:
 - (a) The problem should prompt the user with some *meaningful* text to enter the 3 coordinates (6 non-negative integer numbers in total) which are the coordinates of the vertices of the triangle. How you prompt the user and how you get these 6 values is up to you.
 - (b) Compute and print the coordinates of the center of the inscribed circle to a precision of 2 digits after the decimal place. Include a meaningful message describing what the value is.
 - (c) Compute and print the area of the inscribed circle to a precision of 4 digits after the decimal place. Include a meaningful message describing what the value is.
 - (d) Compute and print the area of the triangle to a precision of 4 digits after the decimal place. Include a meaningful message describing what the value is.
 - (e) Compute and print the area of the triangle excluding the area of the inscribed circle (i.e. the green shaded region) to a precision of 3 digits after the decimal place. Include a meaningful message describing what the value is.
7. If, at any time, the user inputs a non-number where a number is expected, the program should output “Sorry, <input> was not a number.” where <input> is replaced by the actual input of the user.

Some useful information.

- To test if a string is an integer you can use the `isnumeric()` method.

```
isANumber = "245".isnumeric()
```

- The distance d between two Cartesian points $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$ is

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

- The area of a circle with radius r is

$$Area = \pi r^2$$

- The area of a triangle with side lengths a, b, c is

$$Area = \sqrt{s(s-a)(s-b)(s-c)}, \quad s = \frac{(a+b+c)}{2}$$

- Given the Cartesian coordinates of a triangle's vertices $A = (x_A, y_A), B = (x_B, y_B), C = (x_C, y_C)$, the center of the triangle's inscribed circle is $O = (x_O, y_O)$,

$$x_O = \frac{(ax_A + bx_B + cx_C)}{p}, \quad y_O = \frac{(ay_A + by_B + cy_C)}{p}$$

where p is the triangle's perimeter $p = a + b + c$. Note that side a is opposite vertex A ; similarly for b and c (see Figure 2).

- The radius r of a circle inscribed in a triangle with side lengths a, b, c is

$$r = \frac{\sqrt{s(s-a)(s-b)(s-c)}}{s}, \quad s = \frac{(a+b+c)}{2}$$