

# Fundamentals of CS and Programming

## Lab 05: Functions

---

### 1. Absolute value:

Write a function that calculates and prints the absolute value of a given number defined by

$$\text{absolute}(x) = |x|$$

### 2. Circle area:

Define a function that calculates and returns the area of a circle given the radius as input

$$A = \pi r^2$$

### 3. Distance Calculator

Write a function that calculates the distance between two points in the Cartesian coordinate system

$$P1 = x1, y1$$

$$P2 = x2, y2$$

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

### 4. Circle(again):

Define a function that calculates and returns the area of a circle given the center of the circle and a point on the circumference of the circle (**Hint: use distance calculator**)

### 5. Palindrome

A palindrome is a word that reads identically backward and forward, such as level or noon. Write a predicate function `IsPalindrome(string)` that returns true if the string `string` is a palindrome. In addition, design and write a test program that calls `IsPalindrome` to demonstrate that it works.

### 6. GCD

write a function that calculates the gcd of two numbers that are given as parameters

## Case Study: Turtle Graphics

There are many modules in Python that provide very powerful features that we can use in our own programs. Some of these can send email, or fetch web pages. The one we'll look at in this lab allows us to create turtles and get them to draw shapes and patterns.

Turtles are fun, but the real purpose of the chapter is to teach ourselves a little more Python, and to develop our theme of computational thinking.

Turtle graphics is a method of programming “vector” graphics using a relative cursor upon a Cartesian plane. Python has a built in module that supports turtle graphics called the “turtle” module. Importing this module gives you access to all the turtle graphics functions you will need to draw vector graphics on the screen.

### Example 1: Draw a line

One of the simplest things you can do using the turtle module is to draw a line. There are always four steps you need to do in order to use the turtle module:

1. **Import** the turtle module. If we skip this step, there'll be no turtle to control.
2. **Create** a turtle to control.
3. **Draw** things. Do stuff. This will also automatically create the screen.
4. Run `turtle.done()`. (**NOT** `bob.done()`!)

Notice that `turtle.done()` will pause the program. You'll need to close the window in order to continue.

#### Code

```
# Step 1: Make all the "turtle" commands available to us.
import turtle

# Step 2: Create a new turtle. We'll call it "bob"
bob = turtle.Turtle()

# Step 3: Move in the direction Bob's facing for 50 pixels
bob.forward(50)

# Step 4: We're done!
turtle.done()
```

#### Expected Output



### Example 2: Drawing a square.

Lines are boring. We can rotate the turtle in order to draw more interesting figures.

#### Code

```
import turtle

silly = turtle.Turtle()

silly.forward(50)
silly.right(90)    # Rotate clockwise by 90 degrees

silly.forward(50)
silly.right(90)

silly.forward(50)
silly.right(90)

silly.forward(50)
silly.right(90)

turtle.done()
```

#### Expected Output



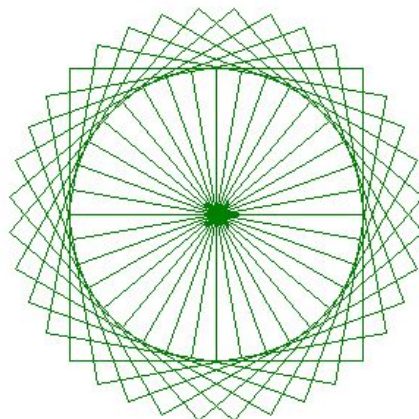
### Exercise

1. Use a loop to draw a square
2. Write a function that draws a square using input parameters

**x, y:** x position and y position

**length:** length of the side of the square

3. Write a program that draws the picture below using the square function written in question number 2



4. Write a function that could plot any regular polygon with equal side length given parameters

- Number of sides
- Side length
- Starting point

(Hint: for a regular polygon the rotation angle is given by the formula

$$360/\text{number\_of\_sides}$$

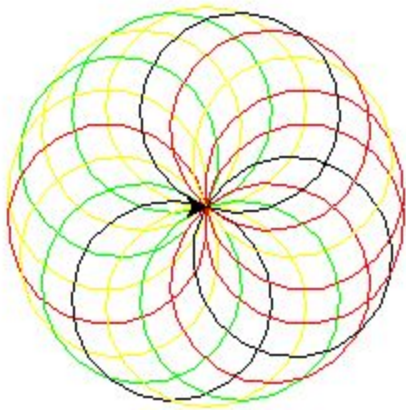
)

5. Turtle has a circle function that takes radius as an input

```
import turtle
t = turtle.Turtle()

t.circle(radius)
```

Write a program using the turtle module that draws the picture below



Use the following function to get random colors for the circles. This function returns random color t that you could give as a parameter for your turtle\_name.color()

```
def get_random_color():
    import random
    color = '#'
    for a in range(0,6):
        r = random.randint(0,14)
        if r >= 10:
            color += chr(ord('a') + (r-10))
        else:
            color += str(r)
    return color
```

6. Modify the above program to fill the circles with different colors

(Hint: `turtle.fill()`)

Call `fill(True)` before drawing the shape you want to fill, and `fill(False)` when done.

