

Course Name	ITD 2313 – Script Programming
Instructor	Andy Tripp
Student Name	Timothy Obinda
Due date	11/02/2025
Grade	Put grade earned here
Grading Comments	Put instructor comments here

## INSTRUCTIONS FOR THE EXERCISE

You should always read the instructions in full. It is best to do a full read through before starting the assignment. Each screenshot needs to be appropriately labeled.

In this instruction, you are given information about a specific project in the text. In some cases, specific instructions for a particular project will be given. There also may be specific test data given. All specific instructions should be followed. When test data is given, screen shots must include that test data, and those results will need to be grabbed via screen shot. All screen shots will go into your submission document.

Some advice, copy this instruction set into your submission document and then put the screen shots under each numbered task. This will take care of making the appropriate labels for you. Each individual book page in the instructions should be in a different screen shot. For any single book page, you may have all the numbered tasks on that page to be in a single screen shot.

**Page 218-219**

**Project # 1**

```
*Circle.py - C:\Users\student\Desktop\Circle.py (3.13.7)*
File Edit Format Run Options Window Help
import math
import turtle

def drawCircle(t, centerX, centerY, radius):
    """
    Draws a circle using a Turtle object.

    Parameters:
        t (turtle.Turtle): The Turtle object used for drawing
        centerX (float): x-coordinate of the circle's center
        centerY (float): y-coordinate of the circle's center
        radius (float): radius of the circle
    """

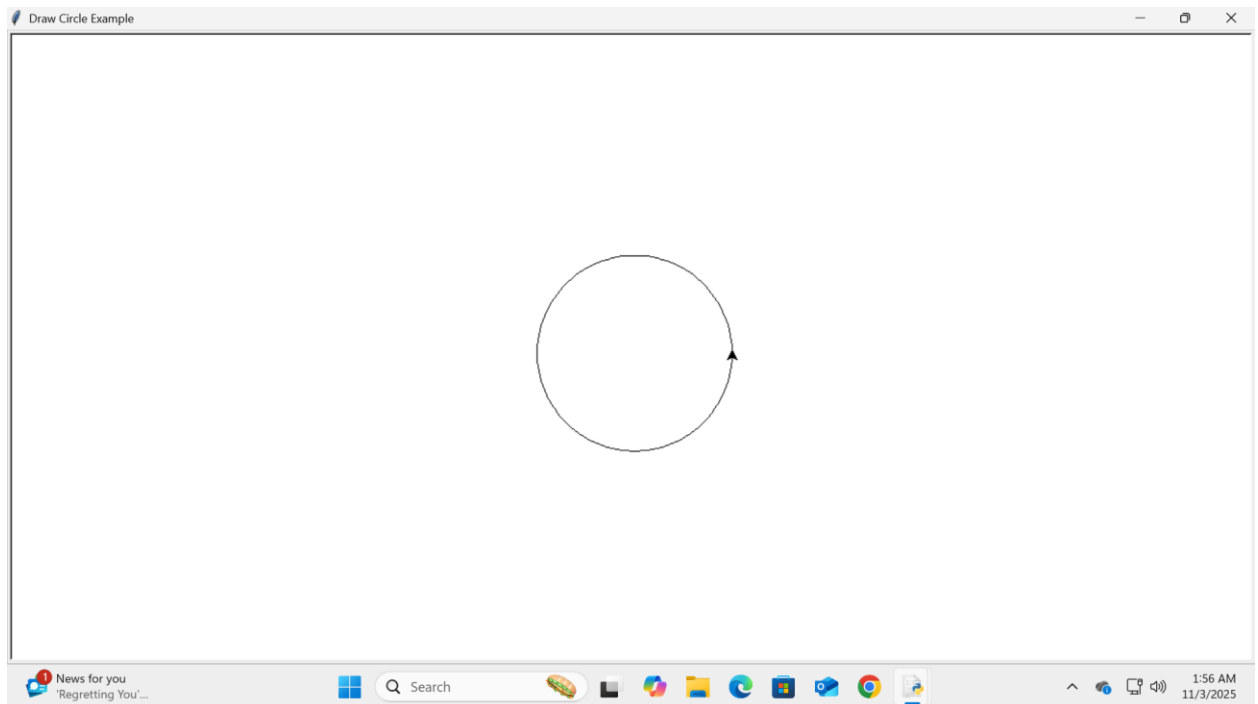
    # Move turtle to the starting point (rightmost point of the circle)
    t.penup()
    t.goto(centerX + radius, centerY)
    t.setheading(90) # Point upward
    t.pendown()

    # Calculate step size for 120 segments (each 3°)
    step_length = 2.0 * math.pi * radius / 120.0

    # Draw the circle
    for _ in range(120):
        t.left(3)
        t.forward(step_length)

# Example test (can be removed when imported)
if __name__ == "__main__":
    wn = turtle.Screen()
    wn.title("Draw Circle Example")

    t = turtle.Turtle()
    t.speed(0)
```



**Special Instructions:**

1. This project is to be done in a Python program file. The file needs to be submitted as part of the submission for this assignment. Place the program file into the zip file that is submitted for this assignment.

**Project # 3**

**Special Instructions:**

1. This project is to be done in a Python program file. The file needs to be submitted as part of the submission for this assignment. Place the program file into the zip file that is submitted for this assignment.

```
koch.py - C:/Users/student/Desktop/koch.py (3.13.7)
File Edit Format Run Options Window Help
drawFractalLine(third, direction, level - 1)

def main():
    # Requested parameters
    width = 200
    height = 200
    size = 150
    level = 4

    screen = turtle.Screen()
    screen.setup(width, height)
    screen.title(f"Koch Snowflake - Level {level}")

    turtle.hideturtle()
    turtle.speed(0)
    turtle.pensize(1)
    turtle.penup()

    # Centering: start at lower-left vertex of an equilateral triangle of side size
    tri_height = math.sqrt(3) / 2 * size
    start_x = -size / 2
    start_y = -tri_height / 3
    turtle.goto(start_x, start_y)
    turtle.pendown()

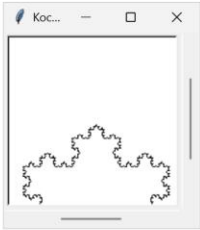
    # Draw the three Koch sides (absolute headings 0, -120, 120)
    drawFractalLine(size, 0, level)
    drawFractalLine(size, -120, level)
    drawFractalLine(size, 120, level)

    turtle.penup()
    turtle.hideturtle()
    turtle.done()

if __name__ == "__main__":
    main()
```

Ln: 60 Col: 0  
1:10 AM  
11/3/2025

```
*IDLE Shell 3.13.7*
File Edit Shell Debug Options Window Help
Python 3.13.7 (tags/v3.13.7:bceec3, Aug 14 2025, 14:15:11) [MSC v.1944 64 bit (AMD64)] on win32
Enter "help" below or click "Help" above for more information.
>>>
===== RESTART: C:/Users/student/Desktop/koch.py =====
```

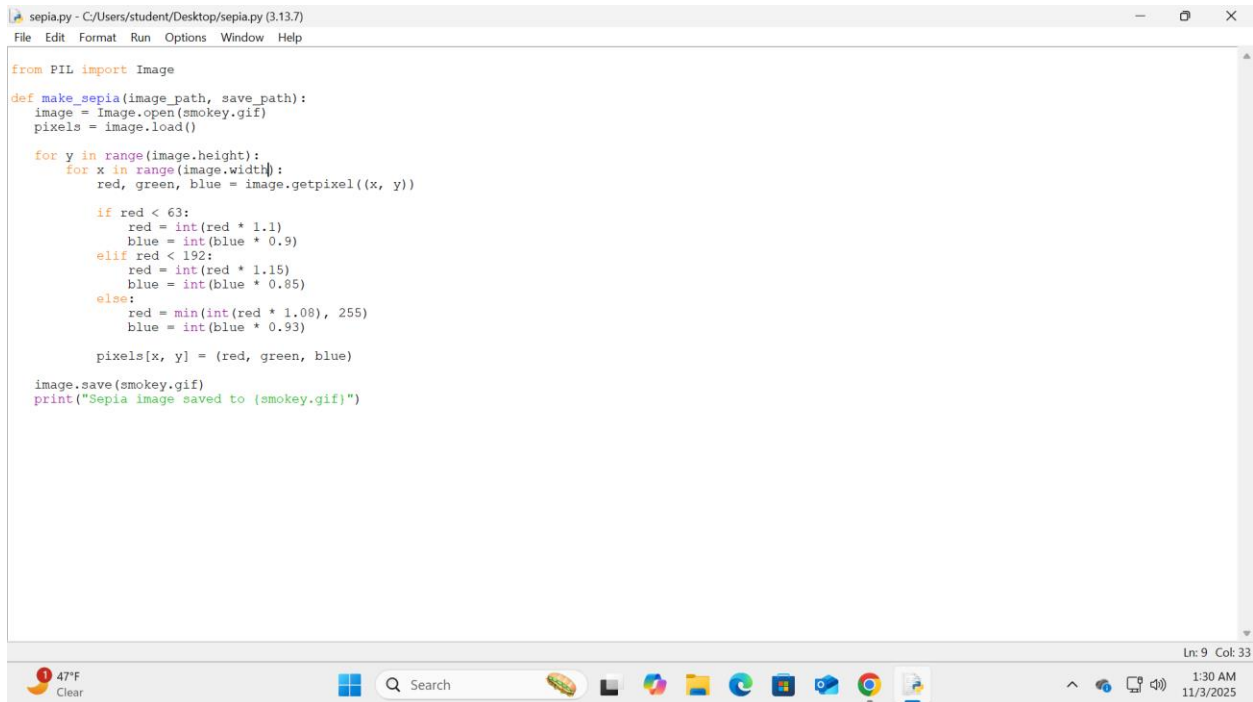


Ln: 5 Col: 0  
1:04 AM  
11/3/2025

## Project # 8

### Special Instructions:

1. This project is to be done in a Python program file. The file needs to be submitted as part of the submission for this assignment. Place the program file into the zip file that is submitted for this assignment.

A screenshot of a Windows desktop environment. At the top, a text editor window titled 'sepia.py - C:/Users/student/Desktop/sepia.py (3.13.7)' is open. The window contains a Python script that uses the PIL library to process an image. The script defines a function 'make\_sepia' that takes an image path and a save path as arguments. It opens the image, loads its pixels, and iterates through each pixel. For each pixel, it checks the red value and applies a color transformation: if red is less than 63, it multiplies red by 1.1 and blue by 0.9; if red is between 63 and 192, it multiplies red by 1.15 and blue by 0.85; otherwise, it sets red to the minimum of (red \* 1.08, 255) and blue to (blue \* 0.93). The transformed pixels are then saved back to the image. The script concludes by saving the image and printing a confirmation message. The bottom of the screen shows the Windows taskbar with a weather widget (47°F, Clear), a search bar, and several application icons. The system clock in the bottom right corner shows '1:30 AM 11/3/2025'.

```
sepia.py - C:/Users/student/Desktop/sepia.py (3.13.7)
File Edit Format Run Options Window Help

from PIL import Image

def make_sepia(image_path, save_path):
    image = Image.open(smokey.gif)
    pixels = image.load()

    for y in range(image.height):
        for x in range(image.width):
            red, green, blue = image.getpixel((x, y))

            if red < 63:
                red = int(red * 1.1)
                blue = int(blue * 0.9)
            elif red < 192:
                red = int(red * 1.15)
                blue = int(blue * 0.85)
            else:
                red = min(int(red * 1.08), 255)
                blue = int(blue * 0.93)

            pixels[x, y] = (red, green, blue)

    image.save(smokey.gif)
    print("Sepia image saved to {smokey.gif}")

Ln: 9 Col: 33
```

47°F Clear    Search    1:30 AM 11/3/2025

