مكتب

Lab Activity

Week 13: Python Turtle Graphics

Duration: 70 minutes — Work in pairs!

Name: _____ **Partner:** _____

Lab Goals - Final Celebration!

By the end of this lab, you will be able to:

- Create beautiful art using Python Turtle Graphics
- Apply loops to make stunning patterns
- Combine all your programming skills in creative ways
- Build confidence as a Python programmer
- Have fun celebrating your learning journey!

Getting Started

1. Extract the given `week13_lab.zip` file to your desktop. This folder contains all the files you need for this lab.

1 Exercise 1: Let's Draw a Colorful Star! (8 minutes)

Open `week13_ex1.py` and type the following code to create a golden star:

```
# Exercise 1: Celebration Star
from turtle import *

speed(5)
color("gold")

# Draw a 5-pointed star
for i in range(5):
    forward(100)
    right(144)

# Add your signature
penup()
goto(0, -120)
color("purple")
write("Made by [Your Name]", align="center", font=("Arial", 12, "bold"))

done()
```

Now modify the code:

- Change the color to your favorite color

- Make the star bigger by changing forward(100) to forward(150)
- Change the number of points (try range(8) with right(135))
- Add your actual name in the signature

Checkpoint: Show your teacher your golden star before moving on!

Celebrate: You just created your first piece of turtle art! Well done!

2 Exercise 2: Complete the Rainbow Octagon (7 minutes)

Open week13_ex2.py and fill in the blanks to create a colorful octagon:

```
# Exercise 2: Rainbow Octagon
from turtle import *

speed(8)

print("Creating a rainbow octagon...")

# Fill the octagon with color
penup()
goto(0, -50)
pendown()
color("lightyellow")
begin_fill()
for i in range(8):
    forward(_______)
    right(45)
_______()  # Complete the fill

# An octagon has 8 sides
for i in range(_______):
    # Set color based on which side we're drawing
    if i == 0:
        color("red")
    elif i == 1:
        color("orange")
    elif i == 2:
        color("yellow")
    elif i == 3:
        color("green")
    elif i == 4:
        color("blue")
    elif i == 5:
        color("purple")
    elif i == 6:
        color("pink")
    else:
        color("cyan")

    forward(80)
    right(_______)  # Hint: 360 divided by 8

# Add a message
penup()
```

```
goto(0, 100)
color("black")
write("Octagon Magic!", align="center", font=("Arial", 16, "normal"))

done()
```

Test your octagon:

- Should have 8 sides with different colors
- Should be filled with light yellow
- Each turn should be 45 degrees

Checkpoint: Rainbow octagon complete? Beautiful work!

Pair Programming: Switch who types every 5 minutes!

3 Exercise 3: Experiment with Turtle Stamps (10 minutes)

Let's create a fun stamp pattern:

Step 1: Open week13_ex3.py. Start with this code:

```
# Exercise 3: Stamp Art
from turtle import *

speed(0)  # Fastest speed
shape("turtle")  # Show turtle shape

# Create a circle of turtle stamps
for i in range(12):
    penup()
    forward(80)
    stamp()
    backward(80)
    right(30)

hideturtle()
done()
```

Step 2: Try these experiments (run after each change):

1. What happens if you add `color("blue")` before the stamp?

   Result: _____

2. What happens if you change the shape to `shape("circle")`?

   Result: _____

3. Can you make each stamp a different color using a list of colors?

   Hint: you can use if conditional statements to change the color based on the loop index.

Tip: The stamp() function leaves a copy of the turtle's shape at the current position!

4 Exercise 4: Fix the Balloon Drawing (8 minutes)

Open week13_ex4.py. This code is meant to draw birthday balloons, but it has some issues.

```
# Exercise 4: Birthday Balloons - Fix me!
from turtle import *

speed(5)

# Draw first balloon
color("red"
beginfill()
circle(50)
end_fill()

# Draw balloon string
right(90)
forward(100

# Move to second balloon position
penup()
goto(100, 0)
pen_down()

# Draw second balloon
color("blue")
begin_fill()
circle(50)
end_fill()

# Draw string
write(90)
forward(100)

# Add birthday message
penup()
goto(50, -150)
write("Happy Coding!", align="center", font=("Arial", 16, "bold"))

done()
```

Checkpoint: Fixed balloons floating properly? Excellent debugging!

5 Exercise 5: Create a Growing Spiral (8 minutes)

Open week13_ex5.py. This code is meant to create a colorful growing spiral, but it has some blanks to fill in.

```
# Exercise 5: Growing Spiral
from turtle import *

speed(0)
pensize(2)

print("Drawing a magical spiral...")

# Draw a spiral that grows bigger
for i in range(30):
```

```
    # Change colors based on the loop number
    if i % 3 == 0:
        color("red")
    elif i % 3 == 1:
        color("blue")
    else:
        color(_______)  # Pick a third color

    # Draw part of the spiral
    forward(_______ * 3)  # Use i to make it grow
    right(_______)  # Turn to create spiral shape

# Add a center decoration
penup()
home()  # Go back to center
pendown()
color("gold")
dot(_______)  # Make a dot in the center

# Add a fun message
penup()
goto(0, -150)
color("black")
write("Spiral Power!", align="center", font=("Arial", 14, "bold"))

done()
```

Your spiral should:

- Start small and grow outward
- Change between 3 colors as it spirals
- Have a golden center dot
- Look like a seashell or galaxy

Hints:

- For the third color, try “green” or “purple”
- Use `i` in the forward command to make it grow
- Try 90 or 91 degrees for the turn angle
- The center dot could be size 20 or 30

Checkpoint: Spiral growing beautifully? Great mathematical art!

6 Exercise 6: Fireworks Display (12 minutes)

Open week13_ex6.py. This code will create a fireworks display using loops.

```
# Exercise 6: Fireworks Celebration
from turtle import *

speed(0)
bgcolor("midnightblue")

```

```
print("Launching fireworks...")

# Firework 1 - Red burst at top
penup()
goto(0, 100)
pendown()
color("red")

# Draw lines radiating out
for i in range(12):
    pendown()
    forward(40)
    dot(10)  # Leave a dot at the end
    backward(40)
    right(30)

# Firework 2 - Yellow burst at top left
penup()
goto(_______, _______)  # Try (-100, 150)
pendown()
color(_______)  # Make it yellow

# Draw the burst
for i in range(12):
    pendown()
    forward(_______)  # Size of burst
    dot(10)
    backward(30)
    right(_______)  # Complete the circle

# Add your own firework 3!
# Move to a new position and create another burst
penup()
goto(_______, _______)  # Choose your position
pendown()
color(_______)  # Choose your color

# Your burst code here:
--------------------------------
--------------------------------
--------------------------------

# Add more fireworks if you want!

# Celebration message
penup()
goto(0, -200)
color("white")
write("Congratulations Python Programmers!", align="center",
      font=("Arial", 18, "bold"))

done()
```

Make your fireworks display special by:

- Adding at least 3 firework bursts total
- Using different colors (gold, orange, pink, cyan, lime)
- Placing them at different heights
- Maybe making some bursts bigger than others

- Adding stars or dots between fireworks

Tip: Copy the burst pattern but change position, color, and size for variety!

Checkpoint: Fireworks lighting up the sky? Time to celebrate your coding journey!

7 Exercise 7: Mini-Challenge - Create Your Signature Pattern (15 minutes)

Create your own unique pattern that represents YOU!

Requirements:

1. Use at least one loop
2. Include at least 3 different colors
3. Add your name or initials somewhere
4. Make it personal - what represents you?

Ideas to inspire you:

- Your favorite sport (football, cricket bat, basketball)
- Your initials in creative style
- Pattern inspired by your favorite thing
- Abstract art that represents your personality
- Something from Pakistani culture (truck art patterns, henna designs)

Starter Template:

```
from turtle import *

speed(5)
bgcolor("white")  # Or choose your background

# Your creative code here!
# Remember:
# - Use loops to avoid repetition
# - Try different shapes and colors
# - Make it unique to you!

# Don't forget to sign your artwork!
penup()
goto(0, -250)
color("black")
write("Created by [Name]", align="center", font=("Arial", 12, "normal"))

done()
```

Pair Programming: Switch who types every 5 minutes!

Celebrate: This is YOUR moment to shine! Show off your creativity and all the skills you've learned!

Checkpoint: Signature pattern complete? Can't wait to see it in the gallery!

8 Lab Summary

Checkpoint: Final checkpoint - make sure you have:

- ☐ Completed all exercises with your partner
- ☐ Created at least one piece of art you're proud of
- ☐ Added your name/signature to your creations
- ☐ Helped your partner when they needed it
- ☐ Had fun celebrating your programming journey!

Reflection - Your Programming Journey (3 minutes)

Rate your growth this semester:

Skill	Week 1	Midway	Now!
Writing Python code	○	○	○
Using loops effectively	○	○	○
Debugging programs	○	○	○
Creating with code	○	○	○

My favorite creation today: _____
One thing I want to create next: _____
Message to next semester's students: _____

Gallery Preparation

Choose your BEST creation from today's lab for the Loop Art Gallery!
Selected piece: _____
Why I'm proud of it: _____

_____

Congratulations, Python Programmer!

You've completed Module 1: Foundations of Programming!
From Scratch blocks to Python masterpieces - what an incredible journey!

Keep coding, keep creating, keep growing!