Python&Math Initiative Project

(PyMath Project)

WP2 – ALGORITHMS AND MATHEMATICS

Session: Day 3, Session 6: Turtle Graphics

Topic: Visual Programming & Geometry

Prof. Dr. Turgay Tugay BİLGİN
Bursa Technical University

Date: 24.11.2025

Contents

CONTENTS

Contents

CONTENTS

01 Intro & Agenda

Session Agenda

Art & Math

Creating complex patterns with Nested Loops.

Geometry

Drawing parallel lines and circles using Loops.

Basic Drawing

Drawing lines and squares.

The Software

Using the Turtle Graphics Environment

New Symbols

Forward, Turn, Home, Clear, Pen.

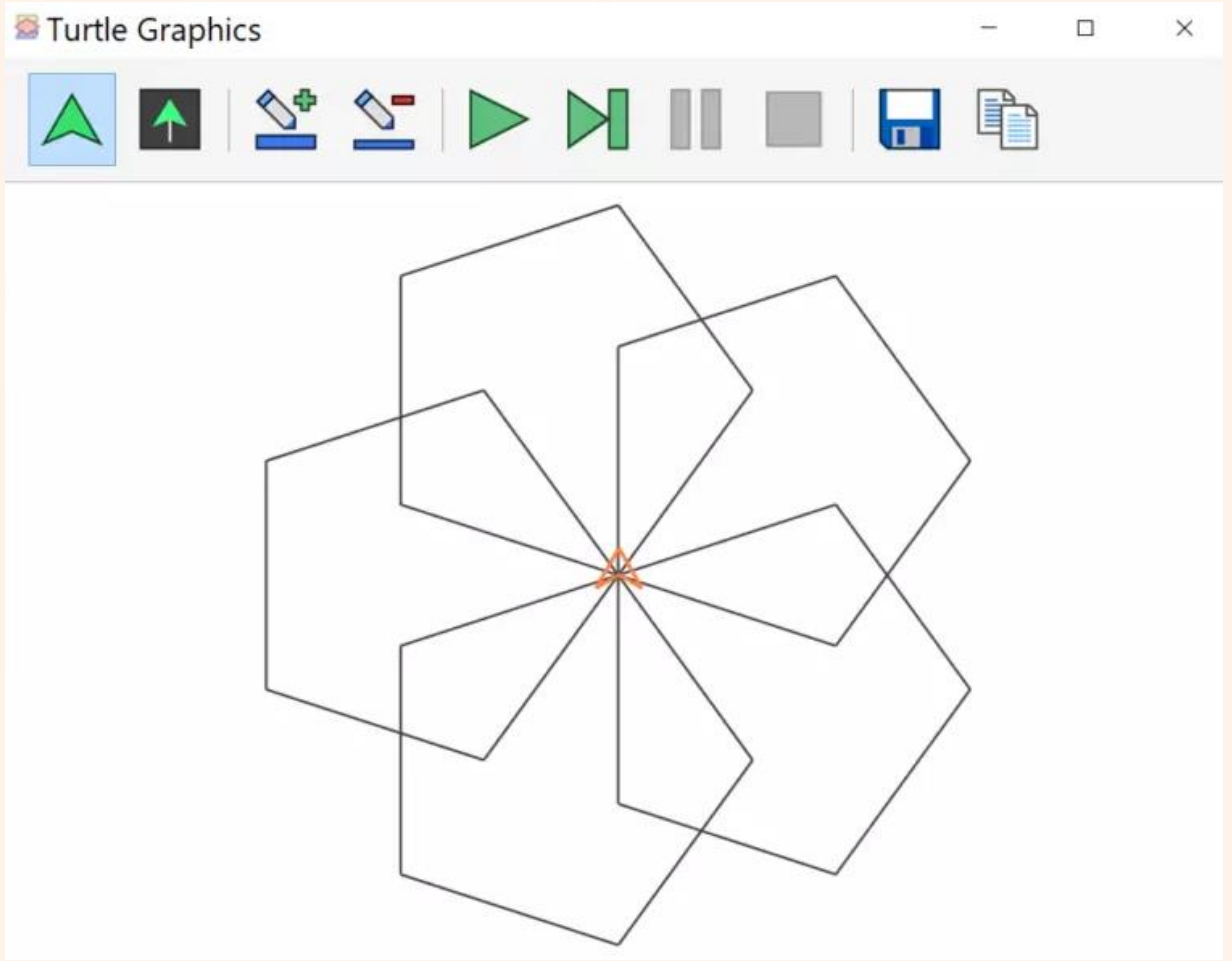
02

Introduction

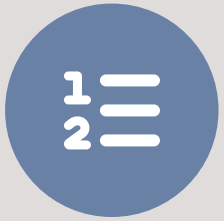
Introduction to Turtle Graphics

Drawing with Code

- Flowgorithm v3 introduced **Turtle Graphics**, an algorithmic drawing software.
- Concept: A "turtle" is a robotic brush on your screen.
- You give it commands like "Move Forward" or "Turn Right".
- As it moves, it draws a line behind it, creating visual art from code.



Why use Turtle Graphics?



Sequential Logic

Students must think step-by-step to create a shape, reinforcing algorithmic thinking.

Geometry

It teaches angles, degrees, shapes, and coordinates in a visual and interactive way.

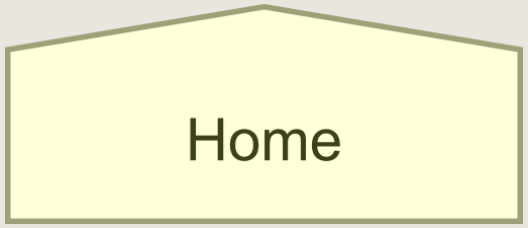
Immediate Feedback

Students instantly see if their logic is correct by looking at the drawing, making debugging intuitive.

03

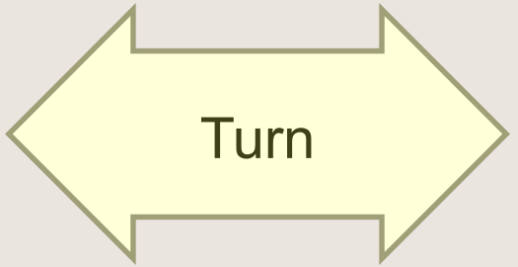
New Symbols

New Flowchart Symbols: Controlling the Turtle



Home

Positions the turtle to the center (0,0) and sets its angle to 90 degrees (facing up).



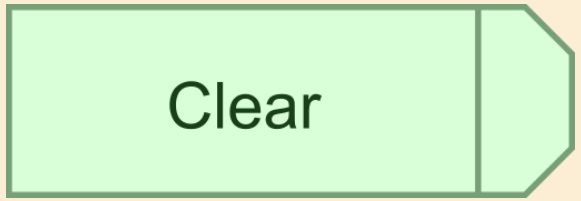
Turn

Rotates the turtle's head left or right by a specified number of degrees.

New Flowchart Symbols: Moving and Clearing

Forward

Moves the turtle. If "Pen Down" is active, it draws a line. Otherwise, it just moves.



Clear

Clears the entire screen and resets the turtle to the Home position.

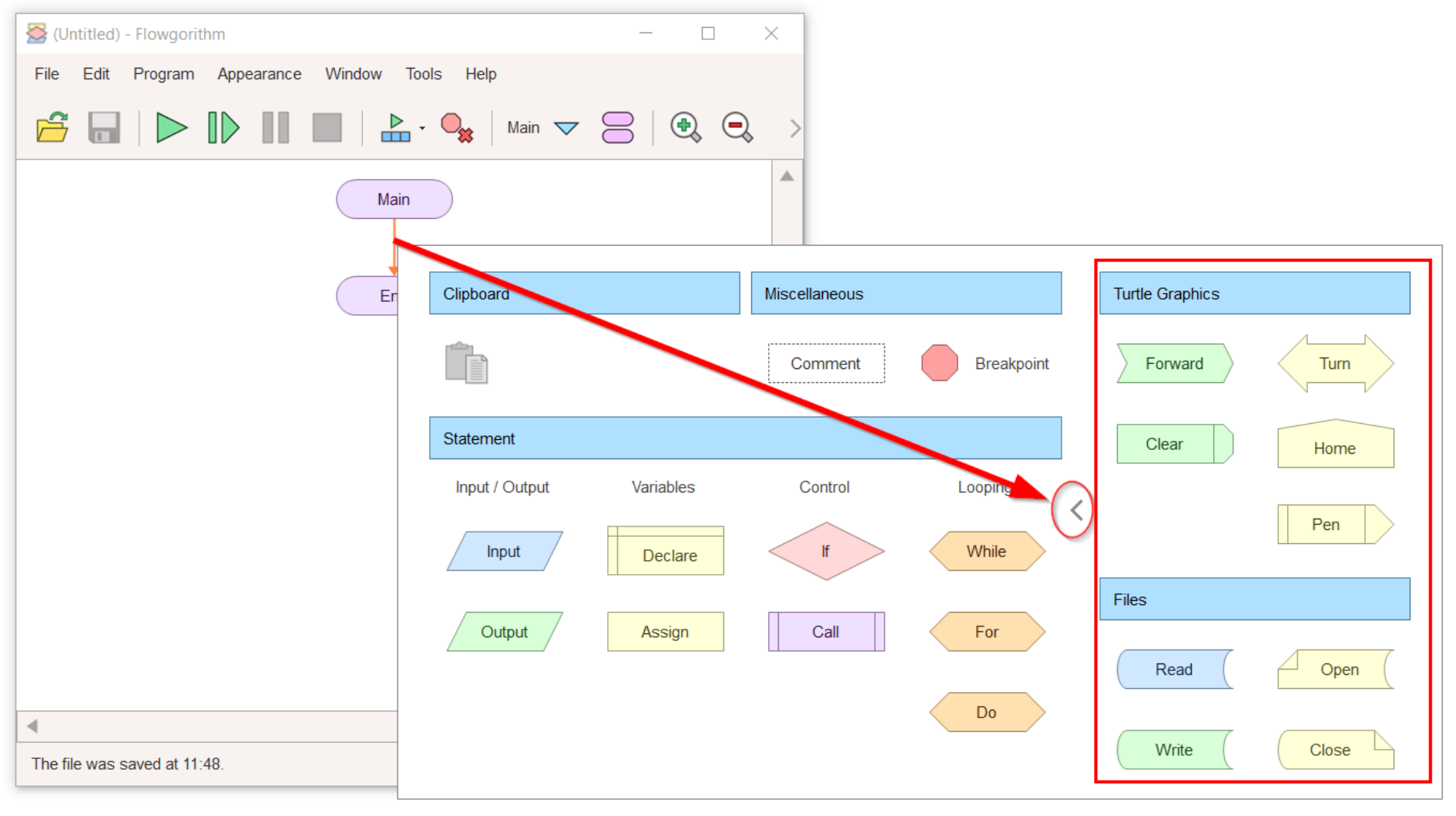
New Flowchart Symbols: Styling

Pen

Controls the color and state of the drawing pen.
The Pen Shape changes the current used by the "turtle" in the Turtle Graphics Window. Flowgorithm supports a total of 10 colors:

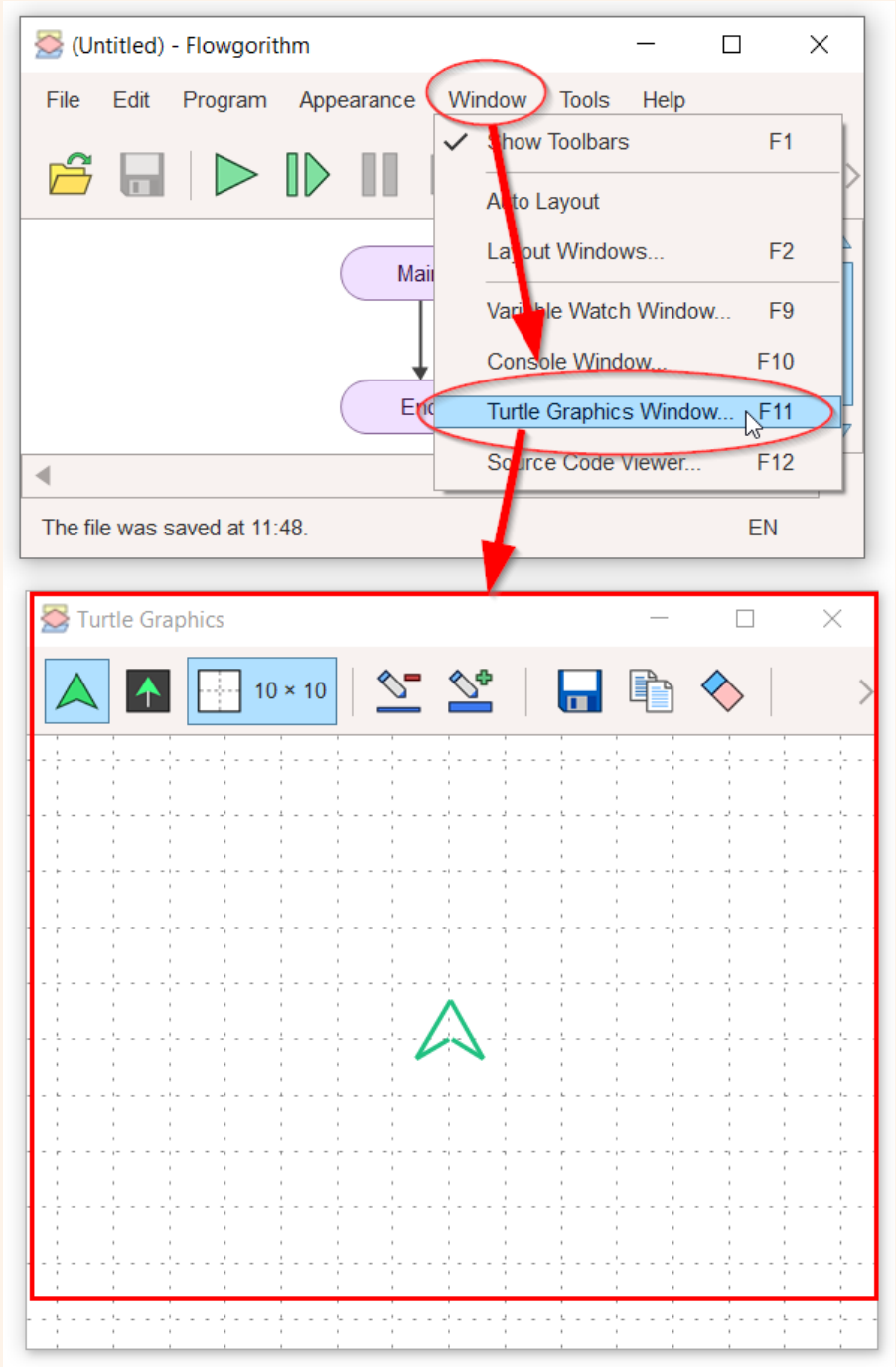
Adding Turtle Symbols

Click the arrow line in your flowchart, then find the **Turtle Graphics** section in the pop-up menu.



04

Interface & Coordinates



The Turtle Graphics Window

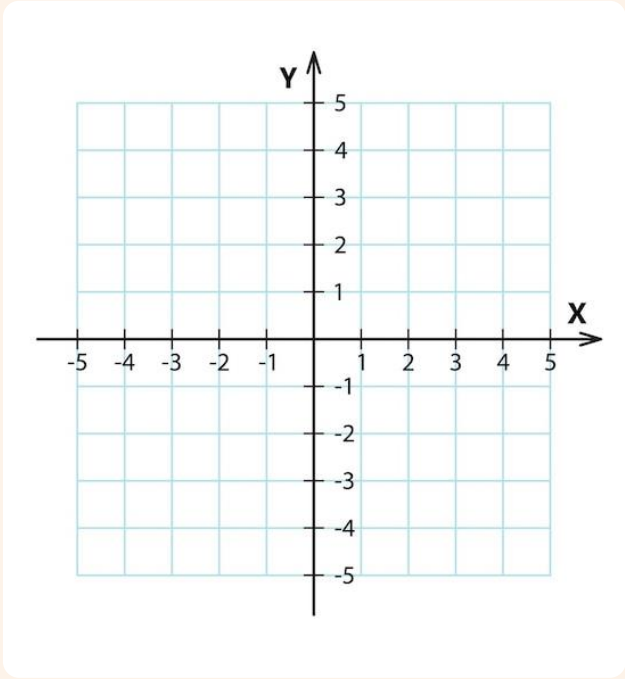
The Canvas

- Go to **Menu -> Window -> Turtle Graphics Window...**
- Or click the Turtle icon in the toolbar.
- The turtle is represented by a small arrowhead showing its direction.

Coordinate System

Math Connection

The window works like a standard Cartesian coordinate system, which is a direct application of mathematical concepts.

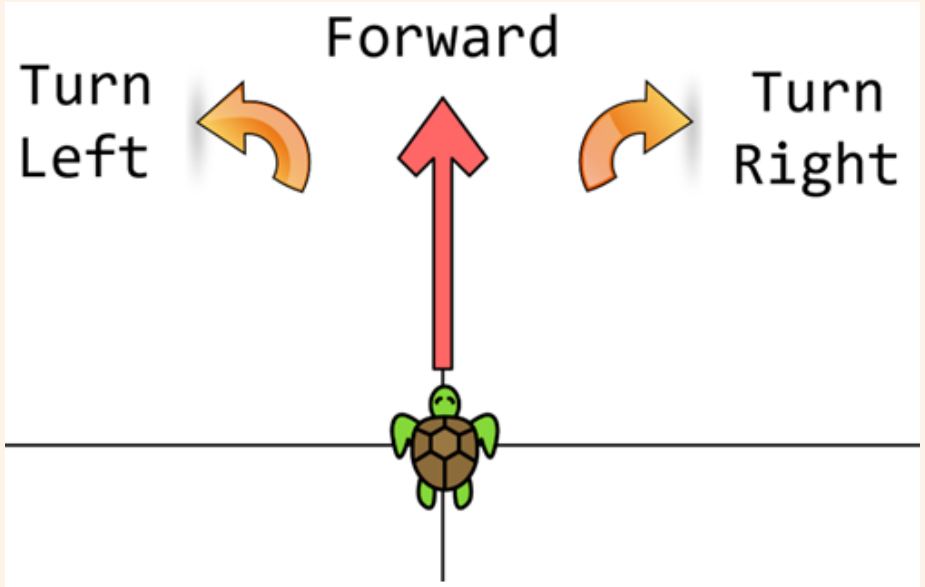


Center (Home): The origin (0, 0) .

X-Axis: Positive is Right, Negative is Left.

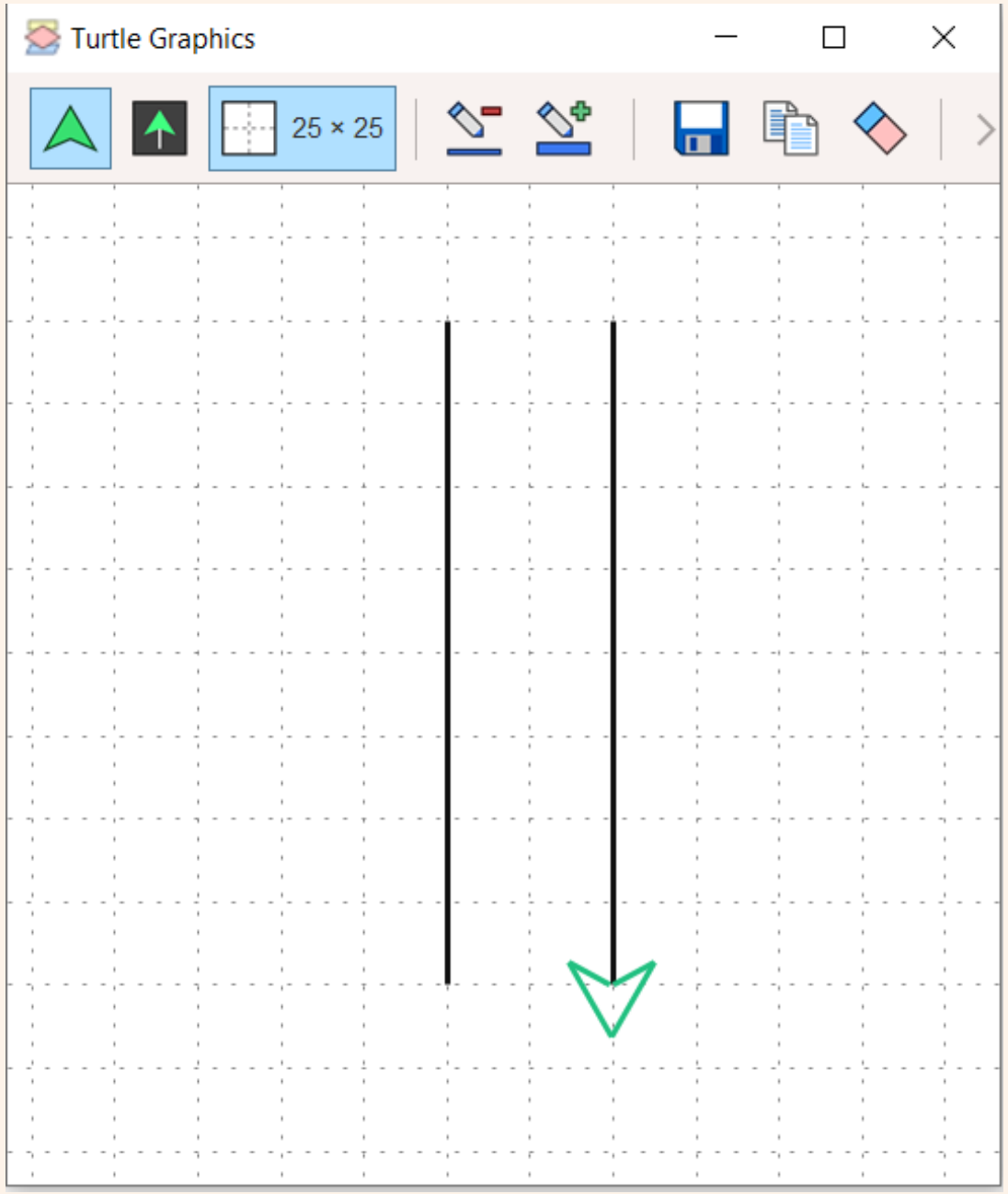
Y-Axis: Positive is Up, Negative is Down.

Forward: A positive number moves in the turtle's facing direction, negative moves backward.



05

Basic Drawings



Activity 1: Parallel Lines

First Drawing

Goal: Draw two parallel lines.

Length: 200 units.

Distance: 50 units between lines.

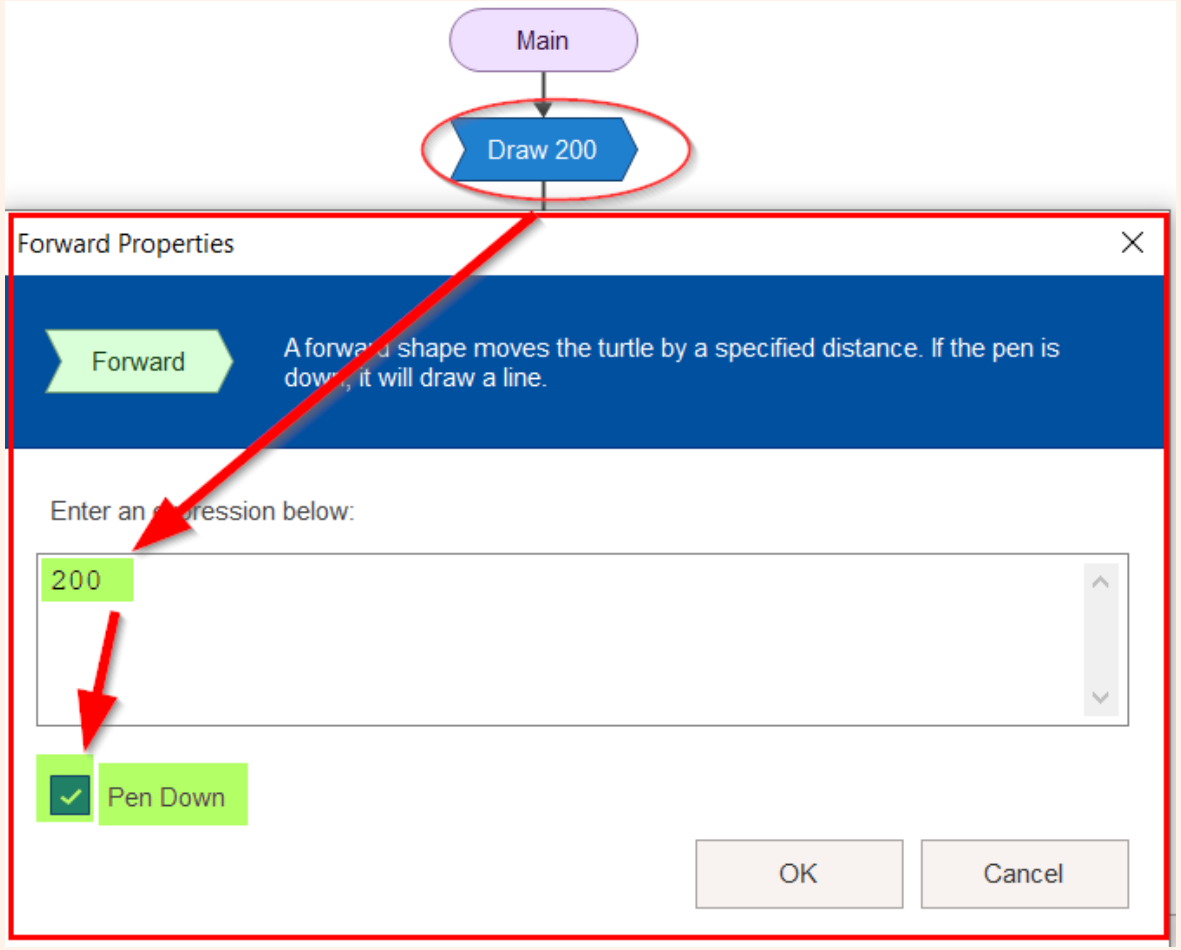
Math Concept: Parallel lines never intersect and are always equidistant.

Algorithm for Parallel Lines

Step-by-Step Logic

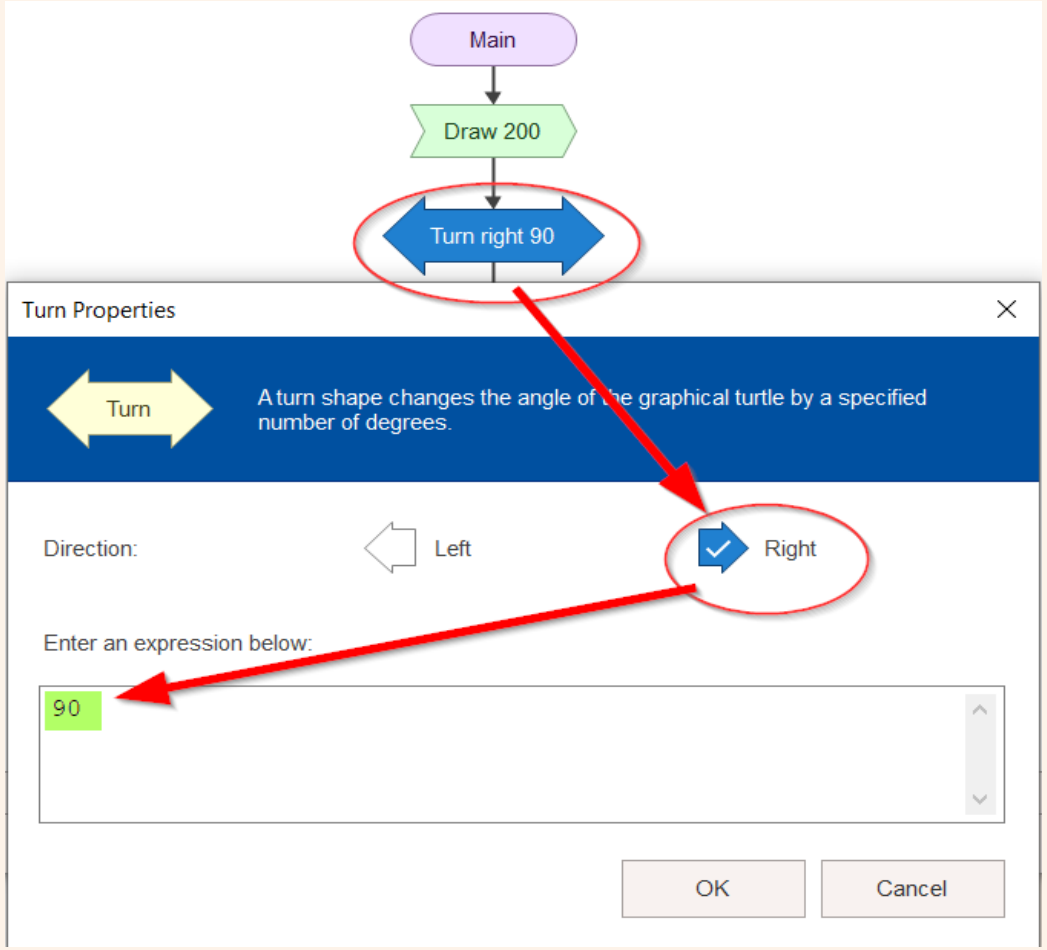
1. Draw a line of 200 units.
2. Turn Right 90 degrees.
3. Lift Pen Up and Move 50 units.
4. Turn Right 90 degrees again.
5. Pen Down and Draw 200 units.
6. Send Turtle Home. (This is optional)

Forward Properties



- **Expression:** Enter `200`.
- **Pen Down:** Check to draw, uncheck to move without drawing.

Turn Properties



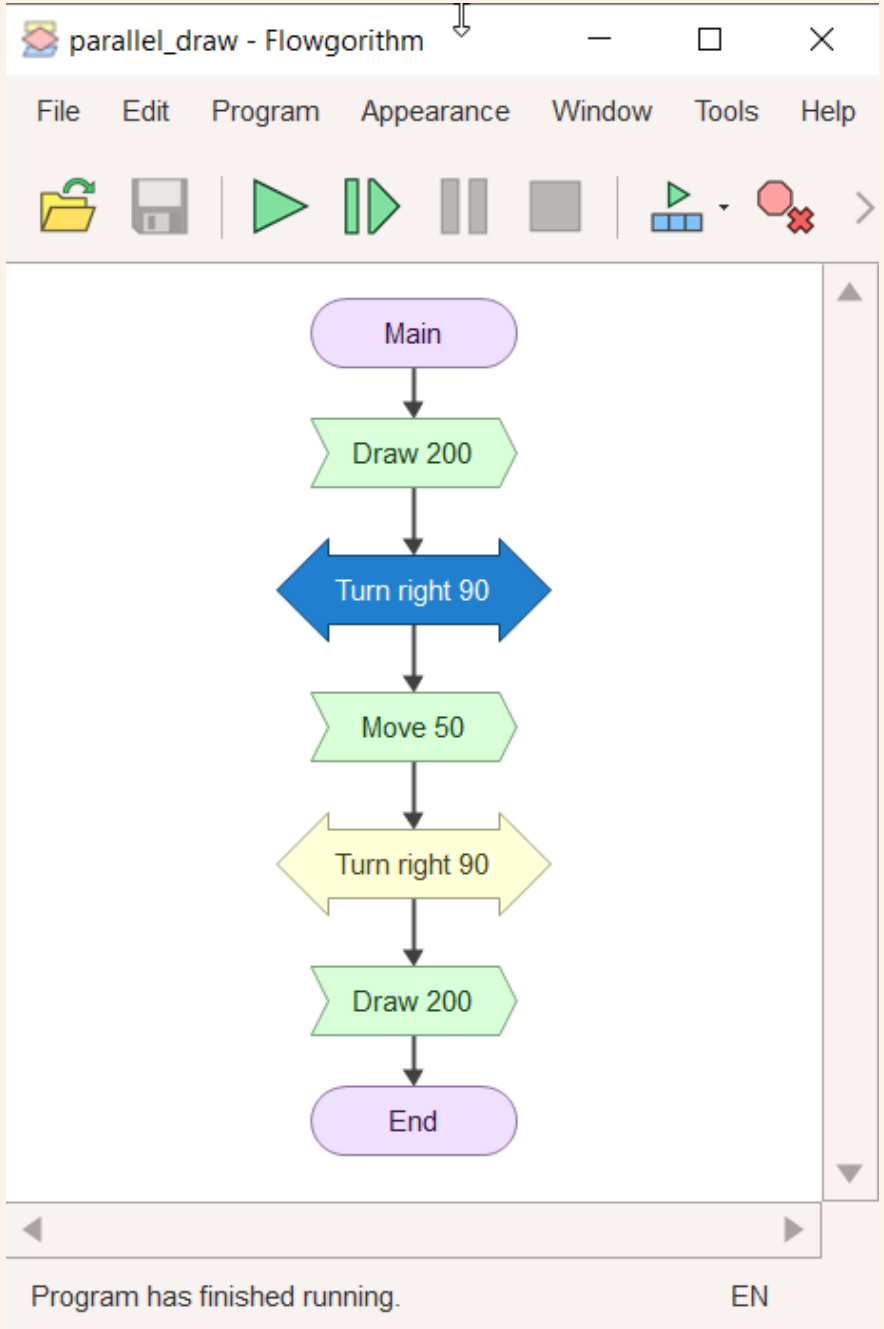
- **Expression:** Enter `90`.
- **Direction:** Select "Right".

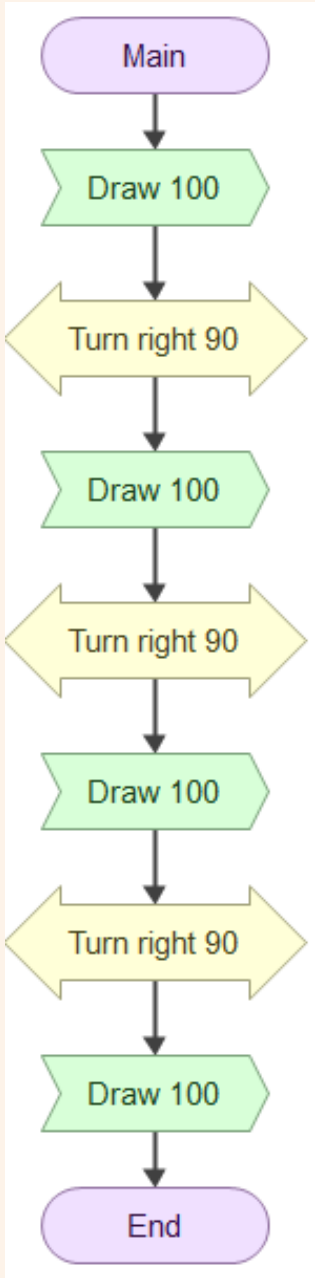
Completed Parallel Lines

This flowchart shows the sequence: Draw -> Turn -> Move (No Pen) -> Turn -> Draw. The result is two parallel lines on the screen.

Key Steps:

Lifting the pen (step 3) is crucial to move to the new drawing position without leaving a trace. Lowering it again (step 5) resumes drawing the second line.





Activity 2: Drawing a Square

Geometric Shapes

Goal: Draw a Square.

Logic: A square has 4 equal sides and 4 right angles (90 degrees).

Approach 1 (Sequential):

Forward 100 -> Turn Right 90 -> Forward 100 -> Turn Right 90 -> Forward 100 -> Turn Right 90 -> Forward 100

06

Drawing using Loops

Optimizing with Loops

Approach 2 (Iterative)

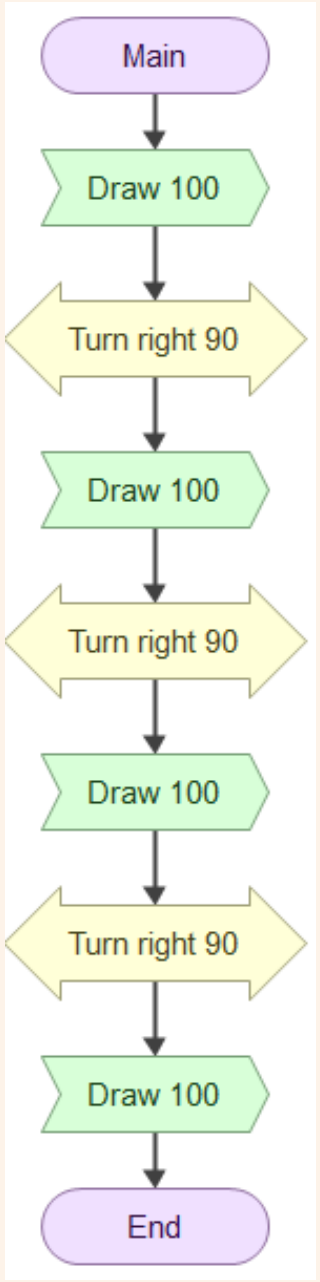
We repeated "Forward" and "Turn" 4 times. This is a perfect candidate for a Loop!

Sequential Code

Forward 100
Turn Right 90
Forward 100
Turn Right 90
Forward 100
Turn Right 90
Forward 100

Loop Code

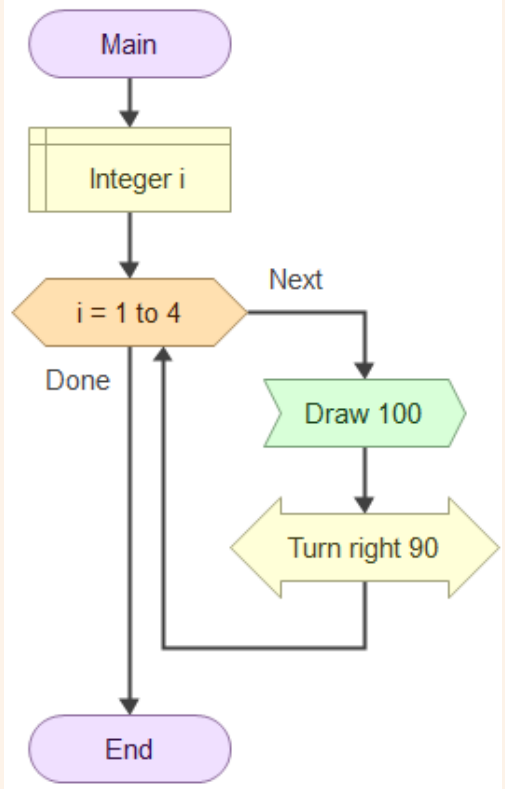
For i = 1 to 4:
Forward 100
Turn Right 90



Optimizing with Loops

Approach 2 (Iterative)

Using loop is much cleaner and easier to modify, for example, to draw a hexagon.



Activity 3: Drawing a Circle

Advanced Geometry

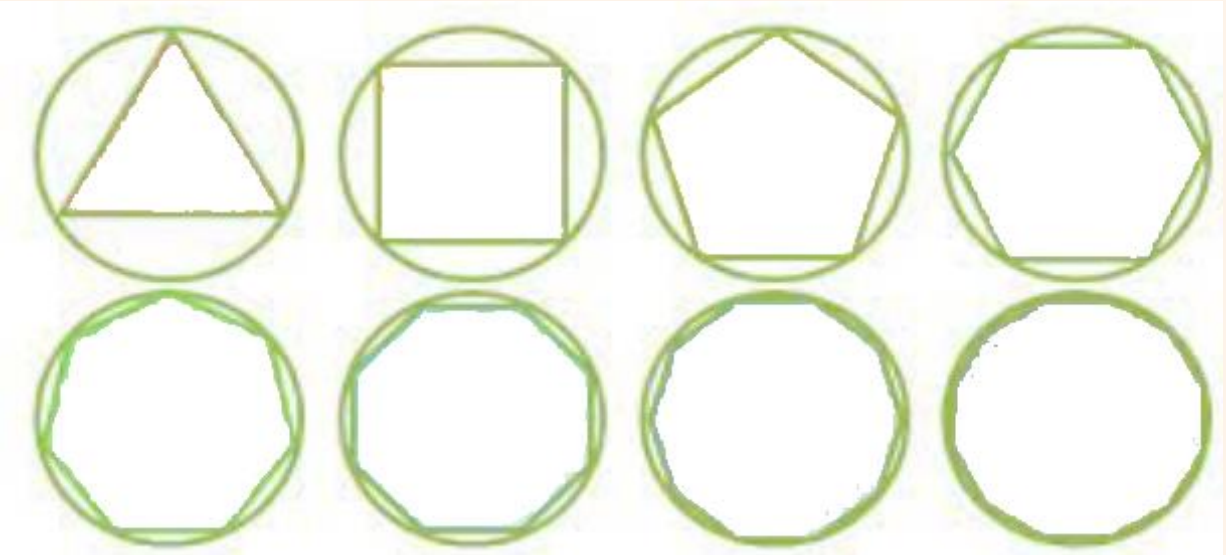
Goal: Draw a Circle.

Problem: There is no "Circle" command.

How do we draw curves with straight lines?

Solution: Approximation.

A circle can be thought of as a polygon with **many** small sides. If we move a tiny bit and turn a tiny bit, many times, it looks like a circle.



07

Circles & Math

Circle Algorithm

The Logic of Approximation

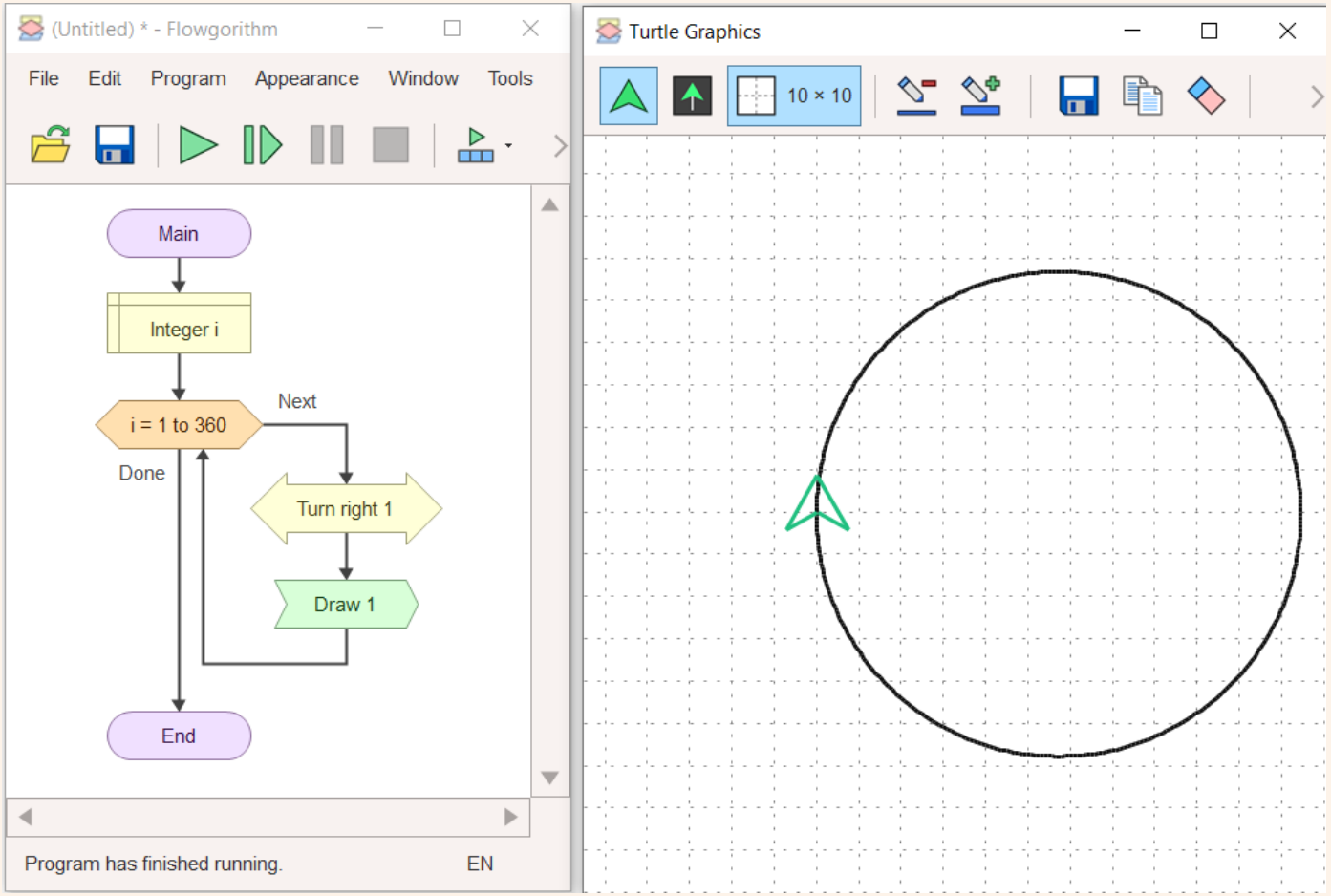
1. Loop: 360 times

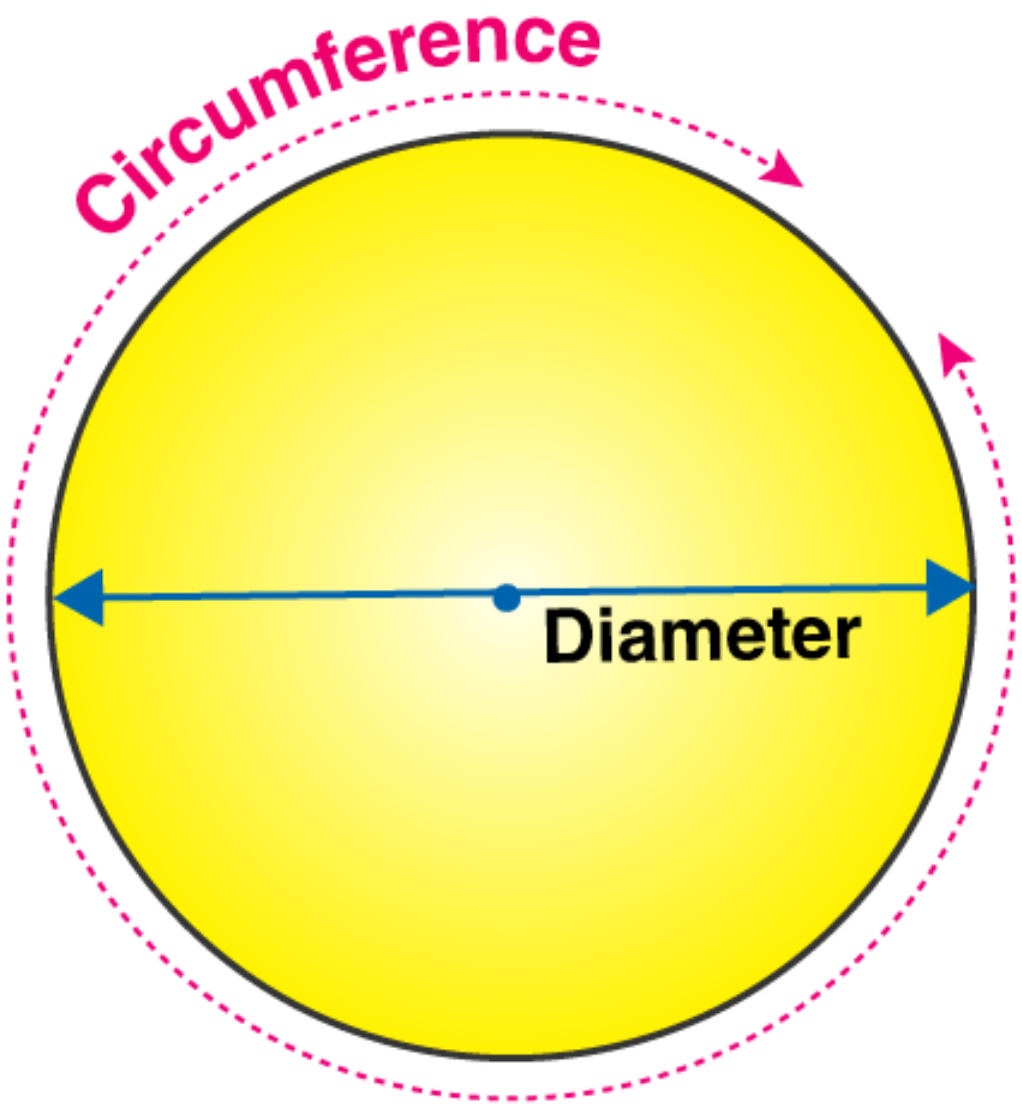
(for 360 degrees in a full circle)

2. Turn Right “1” degree.

3. Move Forward a small amount.

(360 repetitions) x (1 degree) = Full Circle





Circle with Given Radius

Math Application

What if we want a circle with a specific radius `r`?

We know the formula for the circumference C is:
$C = 2\pi r$

In our loop (360 steps), the turtle draws the entire circumference piece by piece.

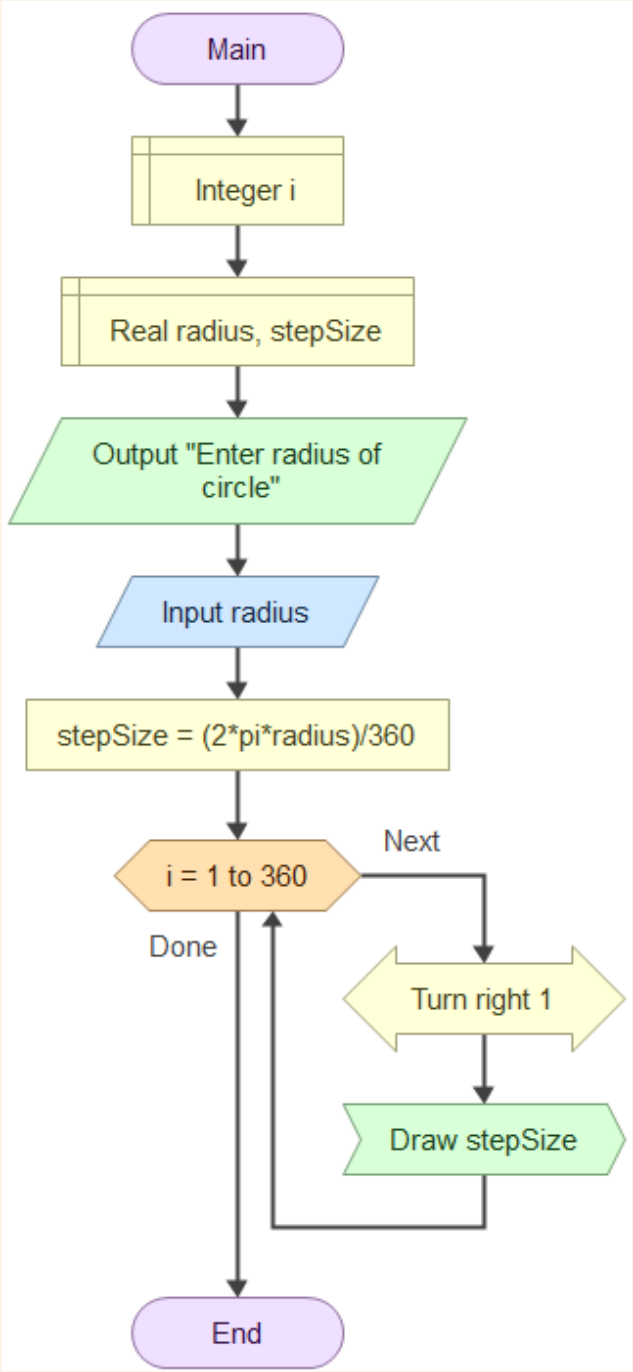
Therefore, each small step must be:

$$stepSize = \frac{2\pi r}{360}$$

Circle with Given Radius Flowchart

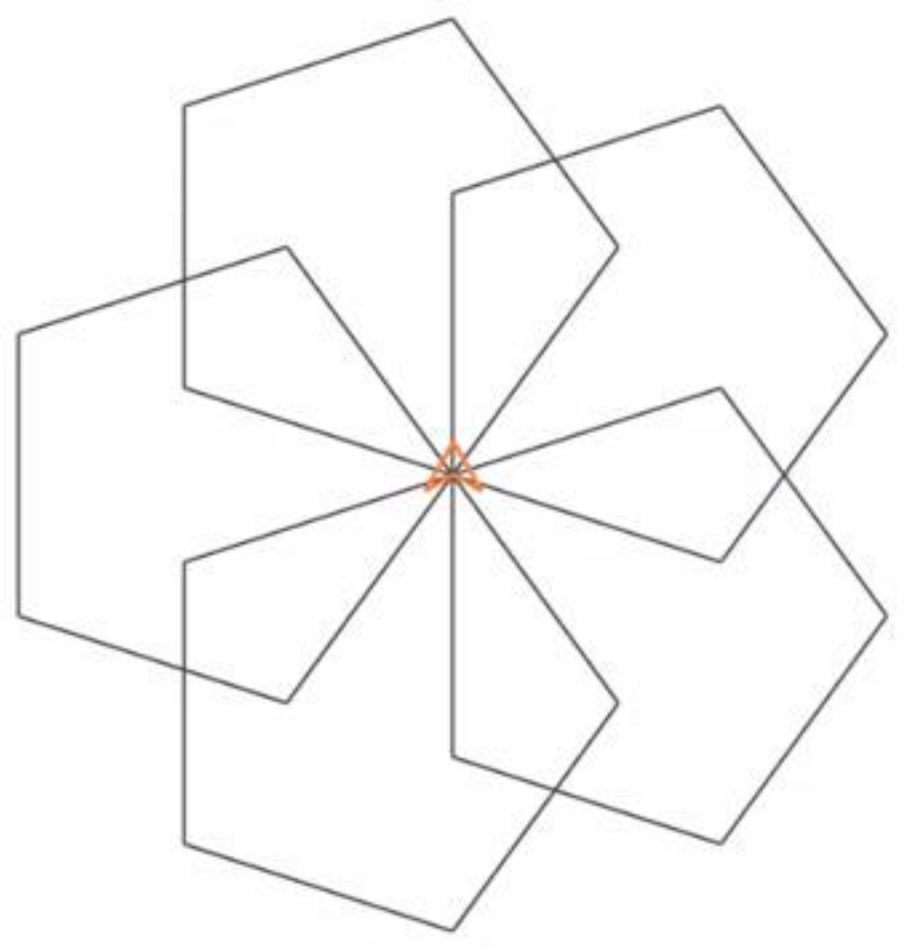
Implementing the formula to draw a circle with a user-defined radius.

1. Input `radius`.
2. Calculate `stepSize = (2 * pi * radius) / 360`.
3. Loop 360 times:
4. Turn Right 1 degree.
5. Forward stepSize.



08

Patterns & Art



Activity 4: Complex Patterns

Art with Nested Loops

We can use Nested Loops (a loop inside another loop) to create spirographs and geometric art.

Inner Loop: Draws a shape (e.g., a Square or Hexagon).

Outer Loop: Rotates the starting angle slightly and repeats the Inner Loop.

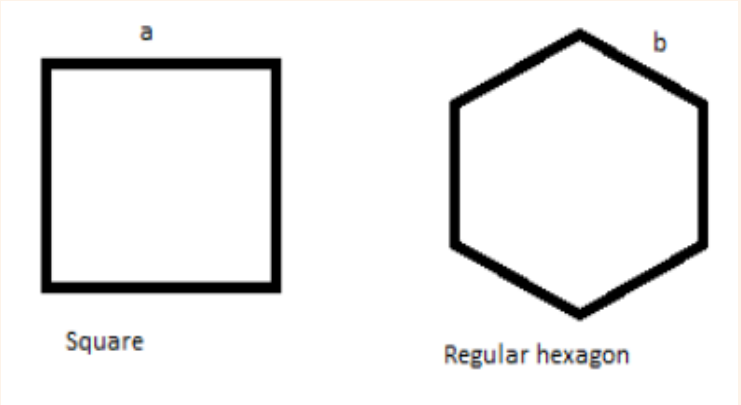
Dynamic Shapes

User Input for Sides

Let's make it dynamic. Ask the user for the number of sides `n` to create different polygons.

Square (n=4)

360 / 4 = 90°



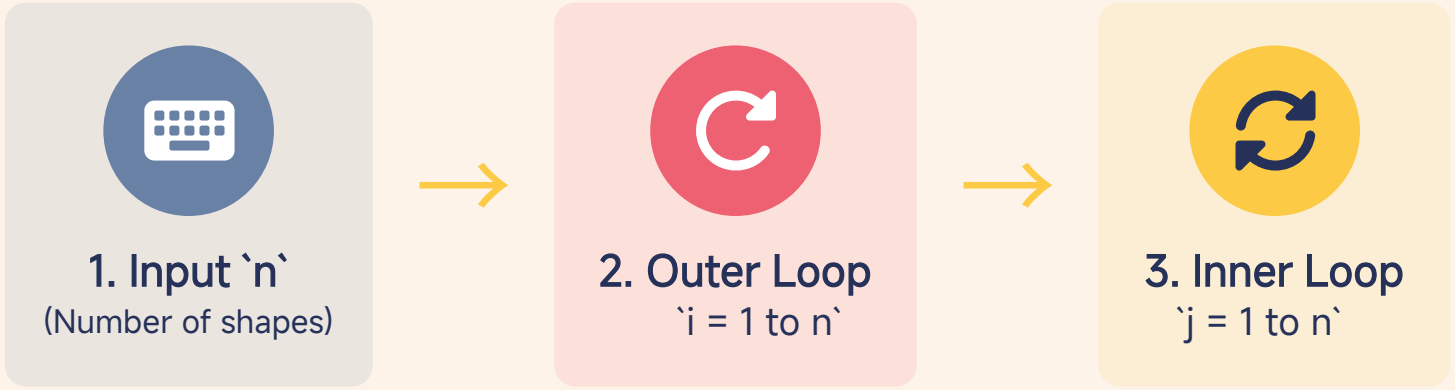
Hexagon (n=6)

360 / 6 = 60°

Formula: Turn Angle = 360 / n

The Pattern Algorithm

Using nested loops to create geometric flowers.



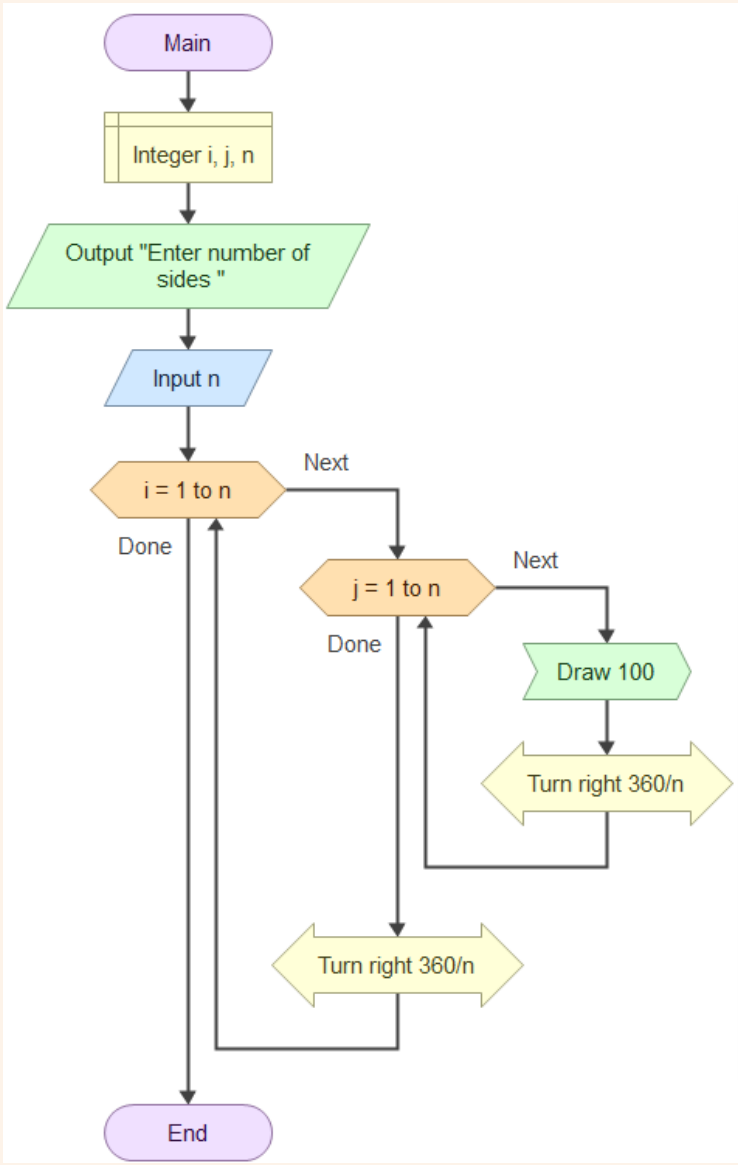
Outer Loop: Turn Right (360/n) → Inner Loop: Forward 100, Turn Right (360/n) →Repeat

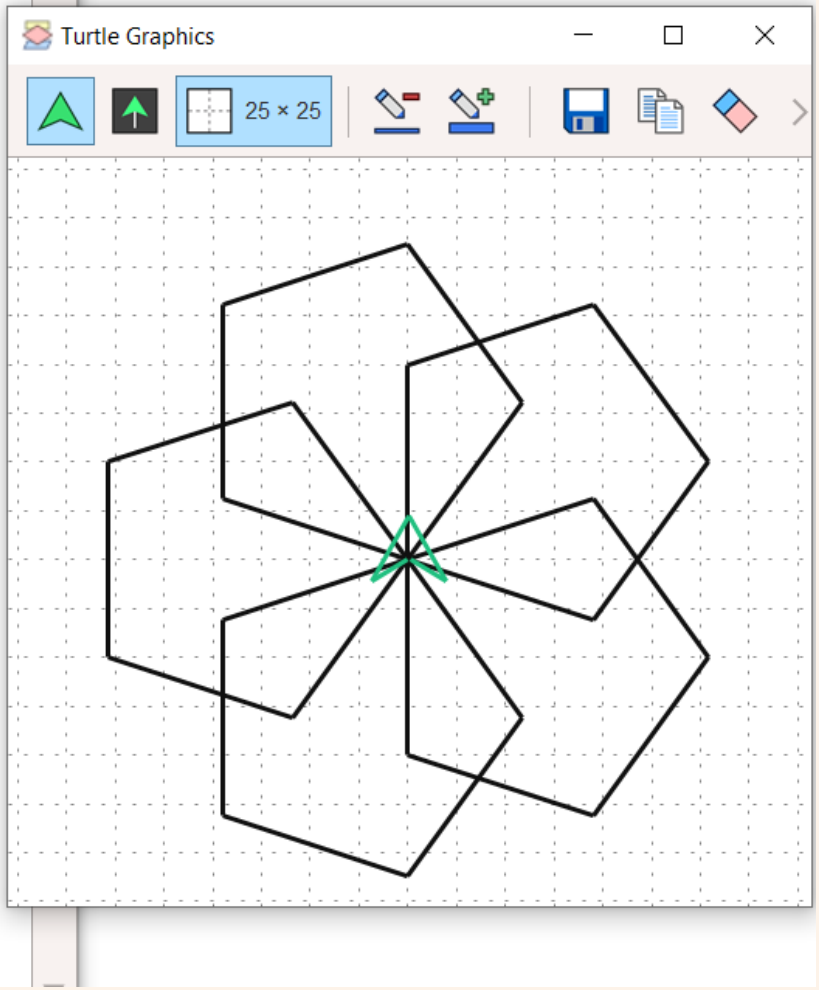
Pattern Result (n=5)

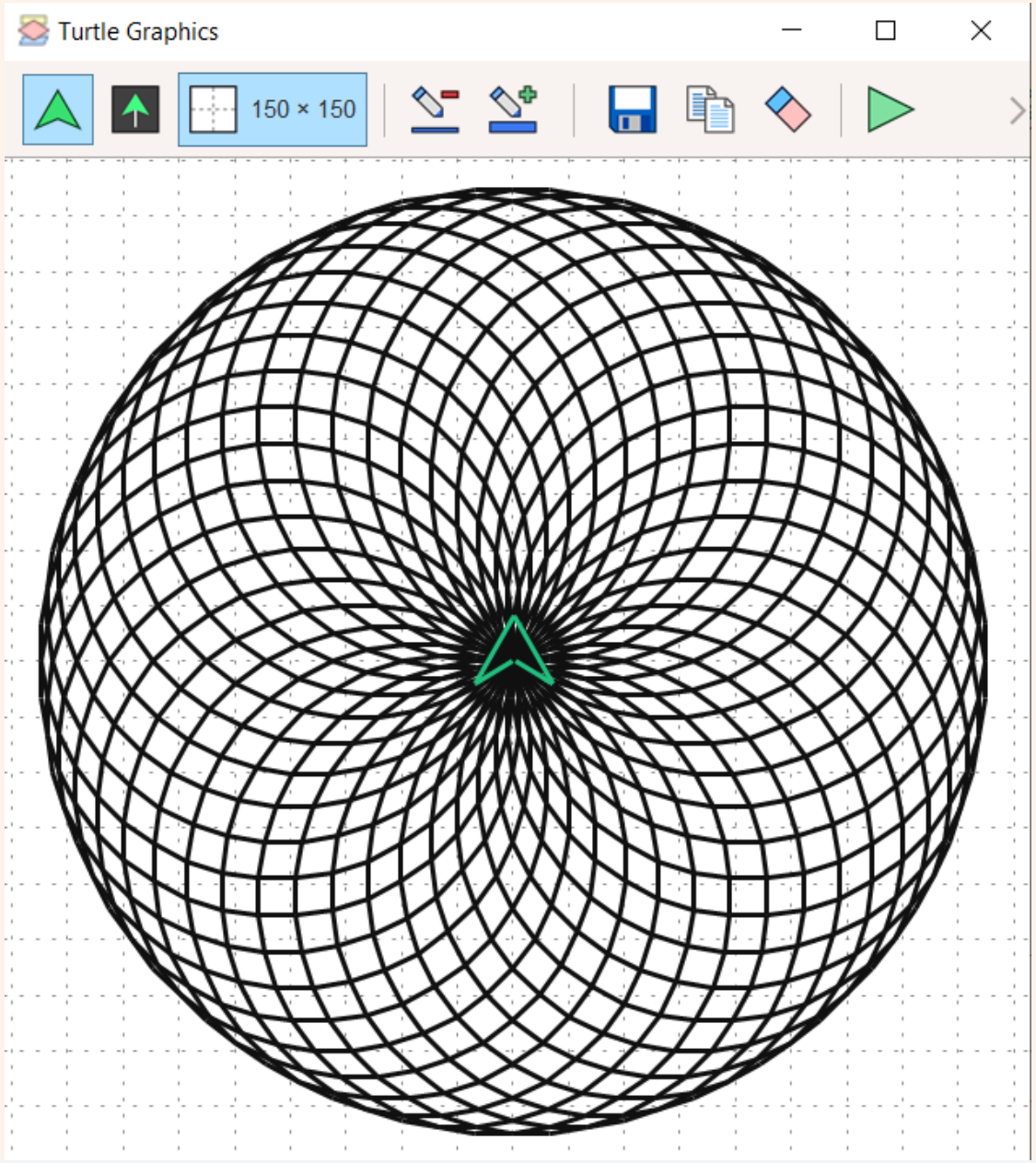
Pentagon Pattern

If the user enters `5`, the inner loop draws a Pentagon. The outer loop draws 5 Pentagons, rotated around a central point.

The result is a beautiful, flower-like geometric pattern.







Pattern Result (n=40)

Complex Art

If the user enters `40`, the inner loop draws a 40-sided polygon (which looks like a circle). The outer loop draws it 40 times.

This creates a complex, beautiful geometric pattern, similar to a Spirograph.

09

Summary & Q&A

Conclusion of Day 3

Wrapping Up

We have covered a lot today! From mastering functions to creating visual art with code, you've built a powerful toolkit for teaching math algorithms.

Session 5: Functions

We mastered Built-in and User-Defined Functions for modular and reusable code.

Session 6: Turtle Graphics

We applied all concepts to create stunning visual art, combining math and creativity.

Assignment: Infinity Symbol

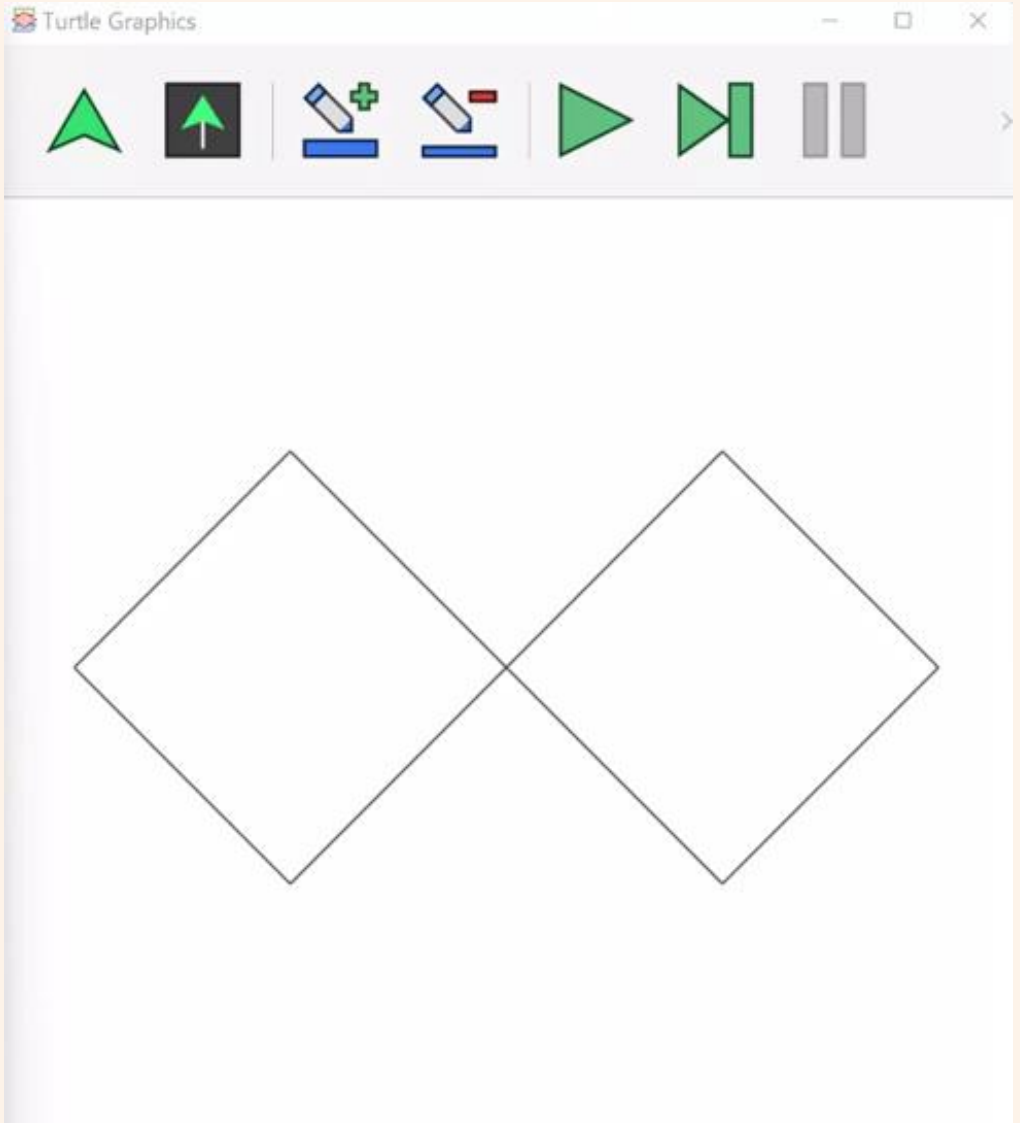
Can you draw this?

Goal: Draw an Infinity Symbol (∞).

> **Hint:**
> It is essentially two circles touching each other. The key is to reverse the turning direction for the second circle.

Logic:

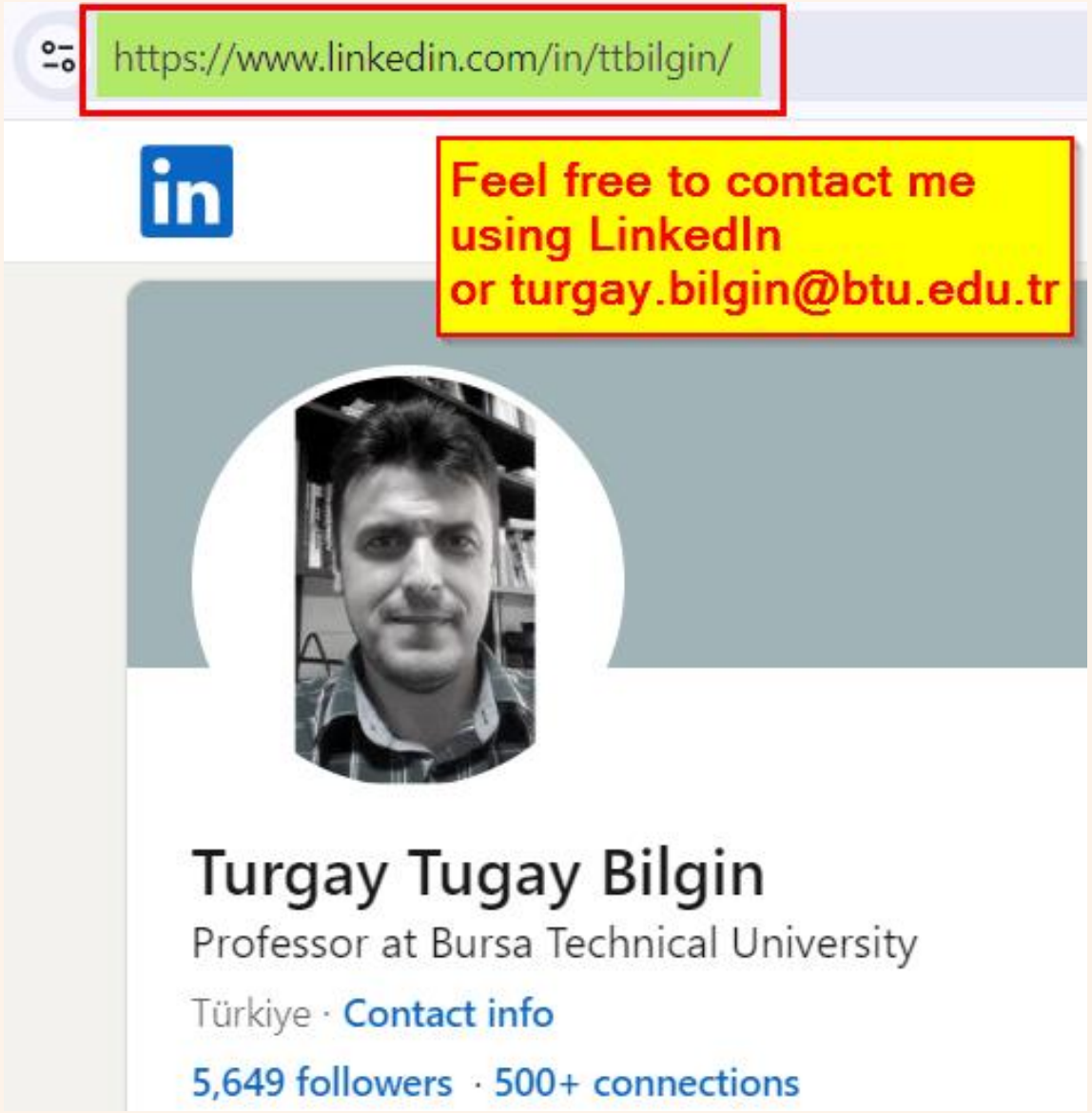
1. Draw a Circle (turning Right).
2. Draw another Circle (turning Left).



Questions?

Any questions about the Turtle window?

Any ideas on how to use this in your geometry classes?



End of Course