**OOPDraw  
Learn the principles of OOP by writing a simple drawing program**

STUDENT WORKBOOK

Created by Richard Pawson

[Creative Commons License](https://creativecommons.org/licenses/by-sa/4.0/)  
This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

# Exercise 1: Using Turtle Graphics

We are initially to make simple drawings of a square and circle, using the ‘Procedural Programming’ paradigm.

You might well have encountered ‘Turtle Graphics’ earlier in your education - instructions are given to an imaginary Turtle to move forward/backward a specified distance, and to rotate (turn) a number of degrees (positive for clockwise, negative for anti-clockwise).

Start by downloading, unzipping, and then opening the OOPDraw project. Find and edit the empty MyDrawings.Draw function, and then add this code into it:

namespace OOPDraw

{

public class MyDrawing

{

public static void Draw()

{

Turtle.PenColor = Color.Blue;

for (int i = 0; i < 4; i++)

{

Turtle.Forward(100);

Turtle.Rotate(90);

}

}

}

}

Run the program and copy a partial screenshot showing the resulting drawing here. Make sure you understand why this generates a square.

Experiment with different values in the Forward(100) function call.

Next we will extract the code for drawing a square into a separate function, passing the sideLength and Colour as paramaters, and also a positionX and position to specify where the square should be started (0,0 being the centre of the screen). Then we will call this function more than once to draw multiple squares.

public static void Draw()

{

Square(0,0,Color.Blue, 100);

Square(100, 0, Color.Red, 50);

}

private static void Square(float positionX, float positionY,   
 Color color, float sideLength)

{

Turtle.X = positionX;

Turtle.Y = positionY;

Turtle.PenColor = color;

for (int i = 0; i < 4; i++)

{

Turtle.Forward(sideLength);

Turtle.Rotate(90);

}

}

Make the changes highlighted above, run the program, and paste in a partial screenshot showing the resulting drawing.

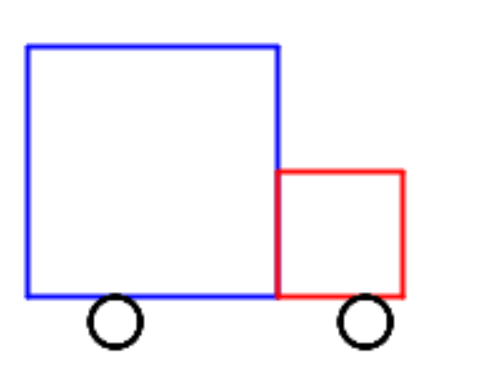
Now create a new function called Circle, which takes four parameters. The first three are the same as for Square, but the last one should be a float named radius.

In Turtle graphics, the simplest way to draw a circle is to draw a polygon with 360 sides - rotating one degree each time. The length of each side can be calculated as:

(float) (2 \* Math.PI \* radius /360)

(The is needed to convert the result of the calculation from a double to a float, which is the type taken by the Forward function).

Within the Draw function, create two small circles positions such that your drawing resembles a truck:



Paste in the code for your Circle function and the Draw function that calls it.