

# Procedural Programming

COMPI400/INFS1609 – Week 1

## What is a program?

Examples:

Input numbers in a spreadsheet, compute means and variances, output a graph.

Input an audio file, compute volume and EQ adjustments, output sound to speakers.

Input joystick controls, compute movement of game pieces, output 3D video.

## What is a program?

A program is a sequence of **instructions** that manipulate **data**.

The data can be numbers, text, images, music, etc...

A program inputs data, rearranges it, and outputs the results.

## Basic Components

**Variables** - containers for storing data.

**Expressions** - ways of combining data to compute new values.

**Statement** - a unit of code, containing one instruction

**Comment** - an explanatory note that has no effect on operation

# Basic Components

**Assignment** - storing data between in a variable

**Conditionals** - testing and branching

**Loops** - doing things multiple times

**Methods** - chunks of re-usable code.

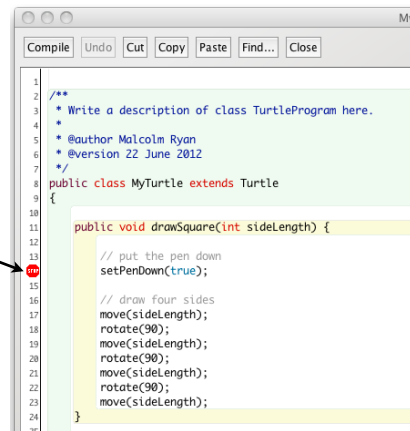
# Advanced Components

**Objects** - structured data with associated methods

**Inheritance** - extending objects to add functionality

## The BlueJ Debugger

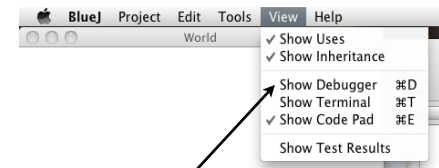
Click in margin to insert a break point.



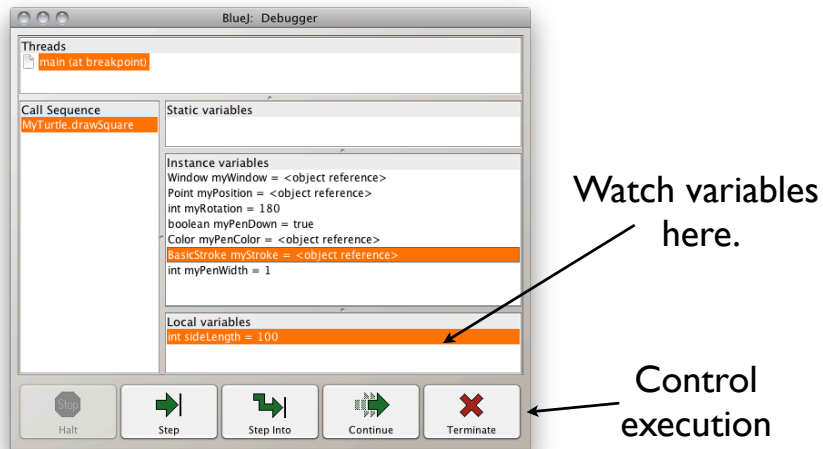
```
1 2
2  /**
3   * Write a description of class TurtleProgram here.
4   *
5   * @author Malcolm Ryan
6   * @version 22 June 2012
7   */
8  public class MyTurtle extends Turtle
9  {
10
11
12
13  public void drawSquare(int sideLength) {
14
15      // put the pen down
16      setPenDown(true);
17
18      // draw four sides
19      move(sideLength);
20      rotate(90);
21      move(sideLength);
22      rotate(90);
23      move(sideLength);
24      rotate(90);
25      move(sideLength);
26  }
```

## The BlueJ Debugger

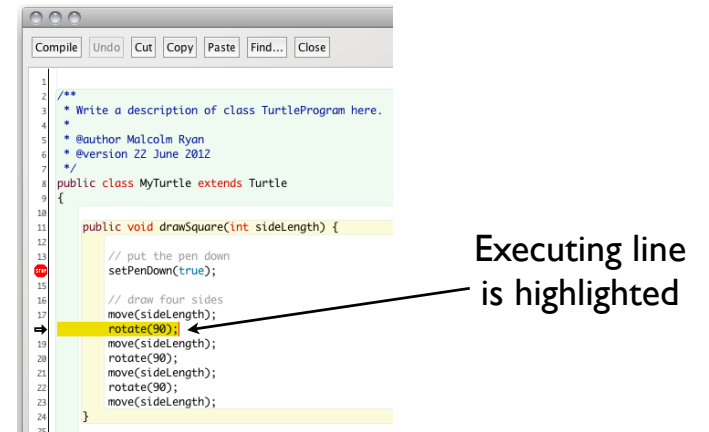
Select "Show Debugger" in View Menu.  
Run code.



# The BlueJ Debugger



# The BlueJ Debugger



## An Example

```
public void drawSquare(int sideLength)
{
    // put the pen down
    setPenDown(true);

    // draw four sides
    move(sideLength);
    rotate(90);
    move(sideLength);
    rotate(90);
    move(sideLength);
    rotate(90);
    move(sideLength);
}

```

## method An Example

```
public void drawSquare(int sideLength)
{
    // put the pen down
    setPenDown(true);

    // draw four sides
    move(sideLength);
    rotate(90);
    move(sideLength);
    rotate(90);
    move(sideLength);
    rotate(90);
    move(sideLength);
}

```

# An Example

parameter

```
public void drawSquare(int sideLength)
{
    // put the pen down
    setPenDown(true);

    // draw four sides
    move(sideLength);
    rotate(90);
    move(sideLength);
    rotate(90);
    move(sideLength);
    rotate(90);
    move(sideLength);
}
```

# An Example

```
public void drawSquare(int sideLength)
{
    // put the pen down
    setPenDown(true);

    // draw four sides
    move(sideLength);
    rotate(90);
    move(sideLength);
    rotate(90);
    move(sideLength);
    rotate(90);
    move(sideLength);
}
```

statements

# An Example

```
public void drawSquare(int sideLength)
{
    // put the pen down
    setPenDown(true);

    // draw four sides
    move(sideLength);
    rotate(90);
    move(sideLength);
    rotate(90);
    move(sideLength);
    rotate(90);
    move(sideLength);
}
```

comments

# Example 2

```
public void drawPolygon(int numberOfSides, int sideLength)
{
    if (numberOfSides < 3)
    {
        System.out.println("Impossible!");
    }
    else
    {
        int angle = 360 / numberOfSides;

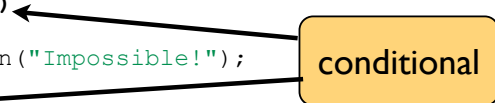
        for (int i = 0; i < numberOfSides; i++)
        {
            move(sideLength);
            rotate(angle);
        }
    }
}
```

variables

## Example 2

```
public void drawPolygon(int numberOfSides, int sideLength)
{
    if (numberOfSides < 3)
    {
        System.out.println("Impossible!");
    }
    else
    {
        int angle = 360 / numberOfSides;

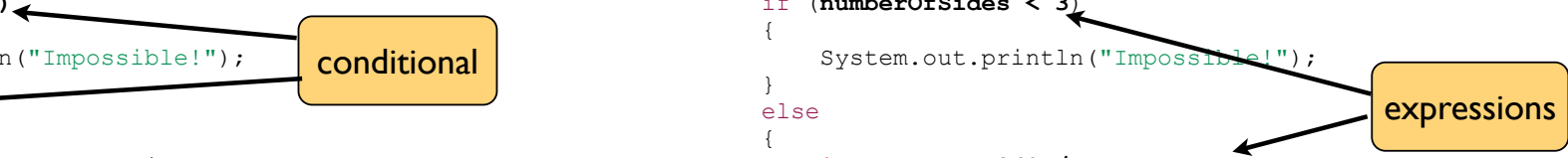
        for (int i = 0; i < numberOfSides; i++)
        {
            move(sideLength);
            rotate(angle);
        }
    }
}
```



## Example 2

```
public void drawPolygon(int numberOfSides, int sideLength)
{
    if (numberOfSides < 3)
    {
        System.out.println("Impossible!");
    }
    else
    {
        int angle = 360 / numberOfSides;

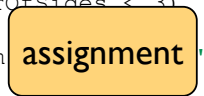
        for (int i = 0; i < numberOfSides; i++)
        {
            move(sideLength);
            rotate(angle);
        }
    }
}
```



## Example 2

```
public void drawPolygon(int numberOfSides, int sideLength)
{
    if (numberOfSides < 3)
    {
        System.out.println("Impossible!");
    }
    else
    {
        int angle = 360 / numberOfSides;

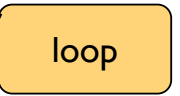
        for (int i = 0; i < numberOfSides; i++)
        {
            move(sideLength);
            rotate(angle);
        }
    }
}
```



## Example 2

```
public void drawPolygon(int numberOfSides, int sideLength)
{
    if (numberOfSides < 3)
    {
        System.out.println("Impossible!");
    }
    else
    {
        int angle = 360 / numberOfSides;

        for (int i = 0; i < numberOfSides; i++)
        {
            move(sideLength);
            rotate(angle);
        }
    }
}
```




## Example 3

```
public void drawSpiral(int length)
{
    while (length > 0)
    {
        move(length);
        rotate(90);
        length = length - 2;
    }
}
```

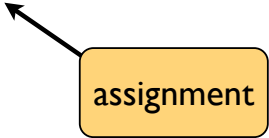
## Example 3

```
public void drawSpiral(int length)
{
    while (length > 0)
    {
        move(length);
        rotate(90);
        length = length - 2;
    }
}
```



## Example 3

```
public void drawSpiral(int length)
{
    while (length > 0)
    {
        move(length);
        rotate(90);
        length = length - 2;
    }
}
```



assignment