

## CSI31 Lecture 13

**Topics:** *Chapter 5. Objects and Graphics*

5.1 Overview

5.2 The object of objects

5.3 Simple graphical programming

**HW# 11** (due date: Wednesday, November 5th):  
draw a house

## 5.1 Overview

### 5.2 The Object of Objects

So far we used built-in Python data types for our programs (int, long, float, str).

Each type

can represent the certain set of data values, and  
has a set of associated operations.

Data – *passive* entities, they are manipulated and combined via *active* operations.

- a **traditional way** to view computation.

Another (**modern**) **approach**:

programs are built using **Object Oriented (OO)** approach.

The basic idea:

we view a complex system as the interaction of simpler **objects**.

**OO objects**

contain data (know staff)

have operations (can do staff)

- in other words, it is a sort of *active* data type.

Objects *interact* by sending each other messages (requests for an object to perform one of its operations).

Example: given on a whiteboard

## 5.3 Simple Graphics Programming

In order to play with graphics, we will use the library written specifically for our book: `graphics.py`.

We need to copy this file from the disk distributed with the book into the folder 'Lib' in the Python's folder.

Then type in the following in the interactive window:

```
>>> win = graphics.GraphWin()
```

-this will create a new object with the name 'win'

you can see that there is a new window called 'Graphics Window'

if we want to close this window, type in:

```
>>> win.close()
```

- the object 'win' is destroyed and we don't see that Graphics Window anymore.

Since we will be working with lots of commands (functions) from graphics library, let's do an alternative import of that library:

```
>>> from graphics import *
```

- it means 'load all the operations/commands from the library module graphics'

- the imported commands become directly available, without the dot-notation.

Now we can work more comfortably:

```
>>> win = GraphWin()
```

When call GraphWin(), by default the size of the window created is 200 pixels × 200 pixels.

**Pixels (picture elements)**– are tiny points on our displays. Each of them has color. By controlling the color of each pixel we can control what is displayed on the screen.

The position of each pixel is a pair (x,y) – x-coordinate and y-coordinate.

upper left corner : (0,0),

lower right corner: (max\_vertical\_resolution, max\_horizontal\_resolution)

I.e. the Origin of our "rectangular coordinate system" is at the top-left corner.

See example program