Objects and Classes

COMP 102

Victoria University of Wellington

Object oriented programming

- Key idea of OO programming
  - program structured into classes of objects.
  - each class specifies a kind of object – eg, the actions it can perform.

- Calling methods in OO languages like java
  - tell an *object* to perform a *method*, passing *arguments*

- Making objects
  - Some objects are predefined.
  - Create objects with bluej:
    - Right-click on class, and select new ……
    - This is how we run programs with BlueJ.
    - not standard, and not a general solution

Objects

Question:

How can a program make new objects?

More Questions:

What is an object anyway?

Why do we need them?

- An object is typically a collection of data with a set of actions it can perform.
  - The objects we have made so far are a bit strange – no data; just actions. (TemperatureConverter, Drawer)

Example of objects

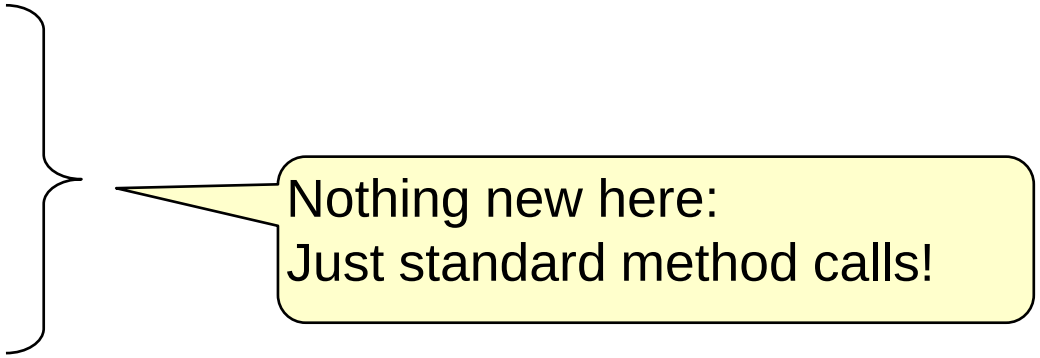
Butterfly program

- Each butterfly is represented by an object which stores the state of the butterfly (position, wing state, direction)
- Butterflies have methods
  - move(double dist) and
  - land()

Using objects

- If the variable bf1 and bf2 contained Butterfly objects, you could do:

```
public void showButterflies(){ // wishful programming
    Butterfly bf1 = ?????
    Butterfly bf2 = ?????
    bf1.move(10);
    bf2.move(20);
    bf1.land();
    bf2.move(20);
    bf1.move(5);
}
```

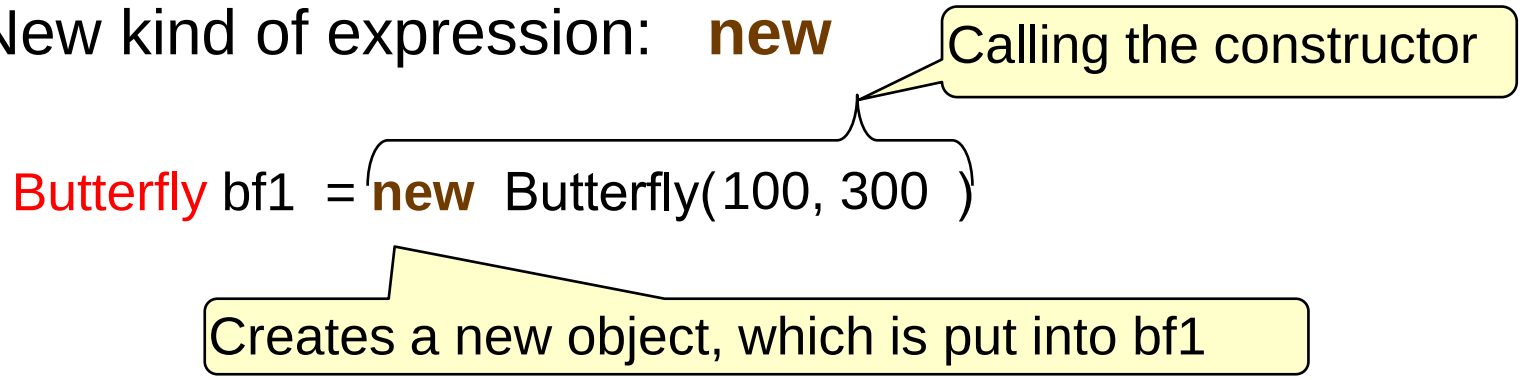


Problem:

How do you get a Butterfly object into the variables?

Creating Objects

- Need to construct new objects:
- New kind of expression: **new**

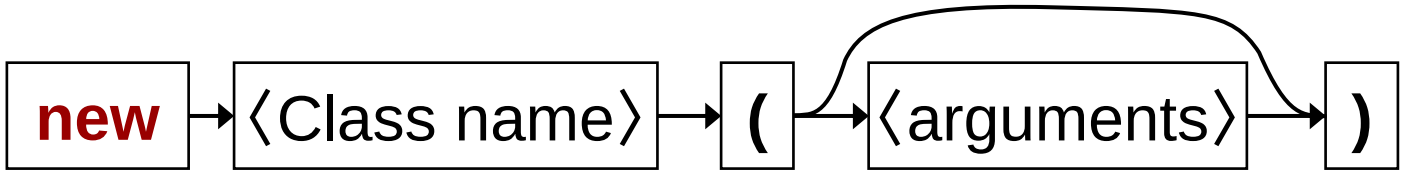


- Constructor calls are like method calls that return a value.
  - have ( )
  - may need to pass arguments
  - returns a value – the new object that was constructed.
- Constructor calls are NOT method calls
  - there is no object to call a method on.
  - must have the keyword **new**
  - name must be the name of the class

Creating Objects: new

```
Butterfly b1 = new Butterfly(100, 300);

UI.setColor( new Color(255, 190, 0) );
```



- Calling a constructor:
  - **new** (a keyword)
  - Butterfly (the type of object to construct)
  - **(…)** (arguments: specifying information needed to construct the new object)
- This is an expression: it returns the new object
  - can put in a variable
  - can use in an enclosing expression or method call

Reading Documentation

- Documentation of a class:
  - Specifies the methods:
    - name
    - type of the return value (or void if no value returned)
    - number and types of the parameters.

      void move (double dist)

      moves the butterfly by dist, in its current direction.

  - Specifies the constructors:
    - number and types of the parameters

      (name is always the name of the class,

      return type is always the class)

      Butterfly(double x, double y)

      requires the initial position of the butterfly

Bluej lets you see the documentation of your classes

Example: Butterfly Grove program

```
public class ButterflyGrove{
   /** A grove of Butterflies which
       fly around and land   */

   public void oneButterfly(){
      Butterfly b1 = new Butterfly(50, 20);
      b1.move(5);
      b1.move(10);
      b1.move(15);
      b1.move(10);
      b1.move(11);
      b1.move(12);
      b1.move(13);
      b1.move(14);
      b1.move(15);
      b1.move(16);
      b1.move(10);
      b1.land();
   }

   public void twoButterflies(){
      Butterfly b1 =   new Butterfly(100, 20);
      b1.move(5);
      b1.move(10);
      b1.move(15);
      double x = 400*Math.random();
      Butterfly b2 = new Butterfly(x, 40);
      b2.move(10);
      b1.move(15);
      b2.move(10);
      b1.move(12);
      b2.move(10);
      b1.move(11);
      b1.move(7);
      b1.land();
      b2.move(20);
      b2.move(25);
      b2.land();
   }
```

Objects are values too:

- Objects can be passed to methods, just like other values.

```
public void Butterflies(){
    Butterfly b1 =   new Butterfly(100, 20);
    Butterfly b2 = new Butterfly(x, 40);
    this.upAndDown(b1);
    this.upAndDown(b2);
}

public void upAndDown(Butterfly b){
    b.move(10);
    b.move(15);
    b.land();
    b.move(15);
    b.move(20);
    b.land();
}
```