Method overloading

Overloading a method

- In Java, methods can have the same name within a class
- This is ***method overloading***
- To ***overload*** a method within a class, any two definitions of the method must have different ***parameter lists***

- methods have different ***parameter lists*** if they have
  - A different <u>number</u> of *parameters*,

  or
  - If the number of *parameters* is the same, then the <u>data type</u> of at least one parameter in the list must differ

- The *signature* of a method consists of the method name and its parameter list.
- Two methods have different signatures if they have either different names or different parameter lists.
- If a method's name is ***overloaded***, then all the methods (with the same name) have different signatures if they have different parameter lists

- Correctly overloaded method

```
public void method1()
public void method1(int x)
public void method1(int x, int y)
public void method1(int a, double b)
```

- Incorrectly overloaded methods

```
public void method1(int a, int b)
public void method1(int x, int y)          ✗

public int method1(int i, double d)
public void method1(int i, double d)
```

- If a method is overloaded, then in the call to that method, the parameter list of the method determines which method is to execute

Methods to determine the larger of two items

Method signatures:

```
public int larger (int x, int y)
public char larger (char c, char d)
public String larger (String str1, String str1)
```

Method calls:

```
larger(2, 3)
larger(‘a’, ‘v’)
larger(“Hello”, “World”)
```

Overloading methods

```
public int min(int n1, int n2).....
```

- `min()` method only works with parameters of type *int*.
- Its **signature** consists of the method name (`min`) and the parameters

  (int n1, int n2)
- What if we want to find the minimum of two *double* values?

Two versions of min

```
// min for integers
public int min(int n1, int n2).....

// min for doubles
public double min(double n1, double n2)...
```

When we call one of these methods, if is clear from the parameters we pass which one we mean.......

Oblong.java

`Oblong.java` has a method `increaseSize()` as shown

```
//method that will increase the dimensions of the oblong object
   public void increaseSize(double widthIn, double heightIn)
   {
      width += widthIn;
      height += heightIn;
   }
```

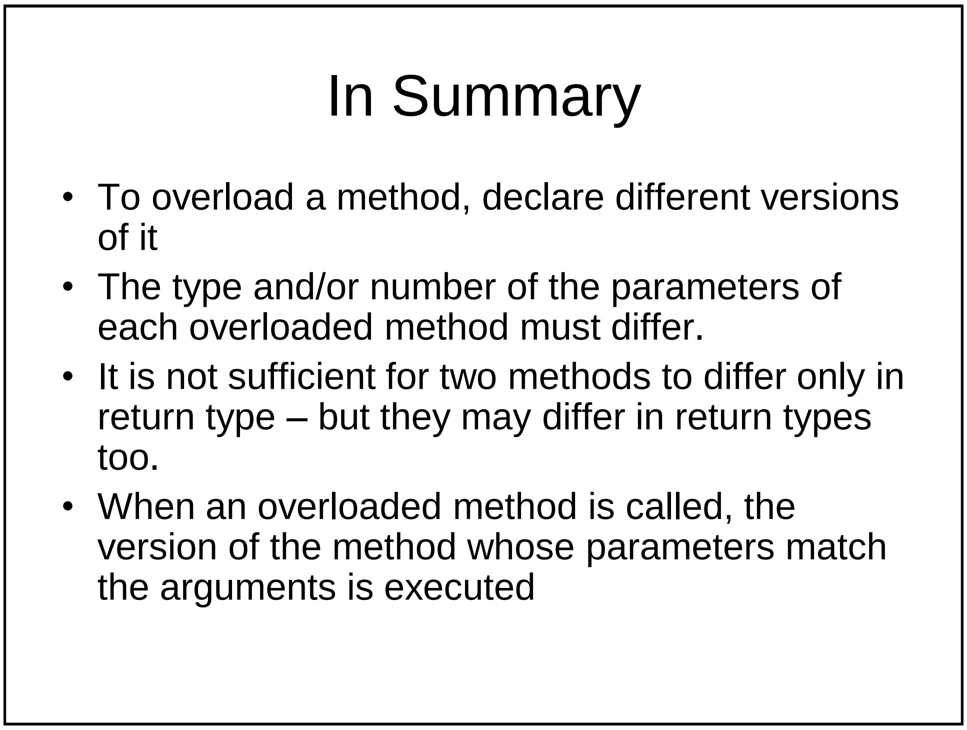
Method call `ob1.increaseSize(10, 14);`

Overload this method so that we can increase
both dimensions by same amount. Valid method call as shown

```
ob1.increaseSize(15);
```

Tips on overloading methods

- Methods that do essentially the same thing should have the same name
- Overloaded methods must differ in their parameter lists
- You cannot have two methods with the same name, same parameter lists and different return types



In Summary

- To overload a method, declare different versions of it
- The type and/or number of the parameters of each overloaded method must differ.
- It is not sufficient for two methods to differ only in return type – but they may differ in return types too.
- When an overloaded method is called, the version of the method whose parameters match the arguments is executed