

CISC 3120

C04: Methods and
Arguments

Hui Chen

Department of Computer & Information Science

CUNY Brooklyn College

Outline

- Recap and issues
- Eclipse and Maven Integration for Eclipse
- Instance variables and methods
- Methods and arguments
- Search solutions online
- Assignments

Class

- Blueprint for
 - What an object knows
 - instance variables, representing states
 - What an object does
 - instance methods, presenting behavior

Example: the Dog class

```
public class Dog {  
    int size; String name;  
    void bark() {  
        if (size > 60) {  
            System.out.println("Woof!");  
        } else if (size > 14) {  
            System.out.println("Ruff!");  
        } else {  
            System.out.println("Yip!");  
        }  
    }  
}
```

```
public class DogTestDrive {  
    public static void main(String[] args) {  
        Dog one = new Dog();  
        one.size = 70;  
        one.bark();  
        Dog two = new Dog();  
        two.size = 10;  
        two.bark();  
    }  
}
```

Passing Parameters

- A method can have parameters
 - Add a method to Dog

```
void bark(int numOfBarks) {  
    for (int i=0; i<numOfBarks; i++) {  
        System.out.println("ruff");  
    }  
}
```

- Caller passes arguments to the method

```
Dog one = new Dog();  
one.bark(3);
```

How about Reference Variables?

```
public class PassByValueDemo {  
    void changeX(int[] x)  
    {  
        System.out.println("changeX: at  
beginning: x[0] = " + x[0]);  
        x[0] = 2;  
        System.out.println("changeX: at the  
end: x[0] = " + x[0]);  
    }  
}
```

```
public class PassByValueDemoTestDrive {  
    public static void main(String[] args) {  
        PassByValueDemo demo = new  
PassByValueDemo();  
        int[] x = {3};  
        System.out.println("main: before  
calling changeX: x[0] = " + x[0]);  
        demo.changeX(x);  
        System.out.println("main: after  
calling changeX: x[0] = " + x[0]);  
    }  
}
```

Return Value

- Method can return a value to the caller

```
public class Calculator {  
    int add(int x, int y) {  
        return x+y;  
    }  
}
```

Encapsulation

- Control access to object states and behavior
 - private, public
 - Compare *GoodDog.java* and *BadDog.java*

Package

- Organize classes into package (thinking C++ namespace)
 - Member without public and private, package accessible

Questions

- Instance variables and methods
- Methods and arguments
- Eclipse and Maven Integration for Eclipse

Search Solution Online

- What should I ask?
- Which one should I trust?
- How do I know I got the solution?
- Am I plagiarizing?

Assignments

- Practice examples to be posted at the class website
- Practice assignments due one week from assignment date (if Thursday, due next Thursday; if Tuesday, due next Tuesday)