COMP 110
More about classes

Darrell Bethea
May 23, 2011
Announcements

- Program 2 due today

- Midterm on Thursday
  - Covers everything we did up through last Thursday
  - Including reading
  - (Ignore parts of sample midterm using Ch. 5)

- Loop worksheet answer sheet mistake
  - #6 should be 43, not 44
  - Updated online
Announcements

- Returning Program 1 today
  - Look carefully over feedback
  - Grading will get more strict

- MANY .jar files were built incorrectly

- New tester for .jar files
  - Test your Program 2 (even if you already submitted)
  - Mandatory starting with Program 3
  - JarChecker.java on the course website
Some Jar Creation Errors

- No Class-Path: line
- No Main-Class: line
- Incorrect class names:
  - Main-Class: bluth
  - Main-Class: Bluth.java
  - Main-Class: Bluth (replace Bluth with the name of your Java class)
- Spaces after class name
- All these detected by JarChecker.jar
Questions?
Classes, Objects, and Methods

- **Class**: a definition of a kind of object

- **Object**: an instance of a class
  - Contains instance variables (data) and methods

- **Methods**
  - Methods that return a value
  - Methods that return nothing
Today in COMP 110

- Local variables and instance variables
- Brief introduction to methods with parameters (aka arguments)
- In-class exercise
Local/Instance variables

- **Instance variables**
  - Declared in a class
  - Confined to the class
    - Can be used anywhere in the class that declares the variable, including inside the class’ methods

- **Local variables**
  - Declared in a method
  - Confined to the method
    - Can only be used inside the method that declares the variable
public class Student {
  public String name;
  public int classYear;
  // ...

  public void printInfo() {
    String info = name + " : " + classYear;
    System.out.println(info);
  }

  public void increaseYear() {
    classYear++;
  }

  public void decreaseYear() {
    classYear--;  
  }
}
public class Student {
    public String name;
    public int classYear;
    // ...

    public void printInfo() {
        String info = name + " : " + classYear;
        System.out.println(info);
    }

    public void increaseYear() {
        classYear++;
        info = "My info string"; // ERROR!!!
    }

    public void decreaseYear() {
        classYear--;
    }
}
public static void main(String[] args) {
    Student jack = new Student();
    jack.name = "Jack Smith";
    jack.major = "Computer Science";

    String info = "Hello there!";
    System.out.println(info);

    System.out.println(jack.name + " is majoring in " + jack.major);

    Student apu = new Student();
    apu.name = "Apu Nahasapeemapetilon";
    apu.major = "Biology";

    System.out.println(apu.name + " is majoring in " + apu.major);
}
Methods with parameters

- Compute the square of this number
  - 5
  - 10
  - 7

- I could give you any number, and you could tell me the square of it
- We can do the same thing with methods
Methods with parameters

- Parameters are used to hold the value that you pass to the method

- Parameters can be used as (local) variables inside the method

```java
public int square(int number) {
    return number * number;
}
```

Parameters go inside parentheses of method header
public class Student
{
    public String name;
    public int classYear;
    // ...

    public void setName(String studentName)
    {
        name = studentName;
    }

    public void setClassYear(int year)
    {
        classYear = year;
    }
}
public static void main(String[] args) {
    Student jack = new Student();
    jack.setName("Jack Smith");
    jack.setClassYear(3);
}
Multiple parameters separated by commas

```java
public double getTotal(double price, double tax)
{
    return price + price * tax;
}
```
Method parameters and arguments

- Order, type, and number of arguments must match parameters specified in method heading
public class SalesComputer
{
    public double getTotal(double price, double tax)
    {
        return price + price * tax;
    }
    // ...

    SalesComputer sc = new SalesComputer();
    double total = sc.getTotal("19.99", Color.RED);
    double total = sc.getTotal(19.99);
    double total = sc.getTotal(19.99, 0.065);
    int price = 50;
    total = sc.getTotal(price, 0.065);
}
Tomorrow

- Even more about classes
- Information Hiding and Encapsulation
In-class exercise