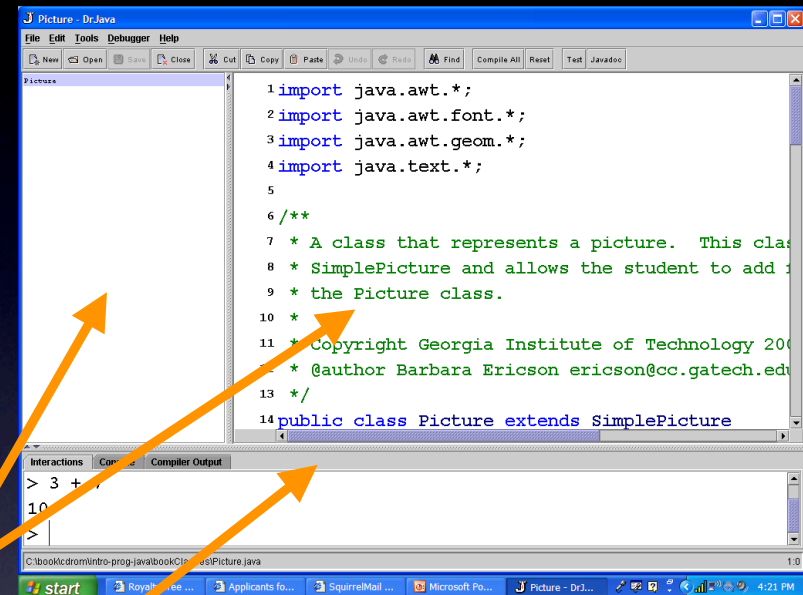# Introduction to Java

CS 1124, Media Computation

November 10, 2008

Steve Harrison

DrJava?

- DrJava is a free integrated development environment for doing Java programming
  - From Rice University
  - It is written in Java
- It has several window panes in it
  - For creating programs (definitions pane)
  - For trying out Java code (interactions pane)
  - Listing of open files (files pane)
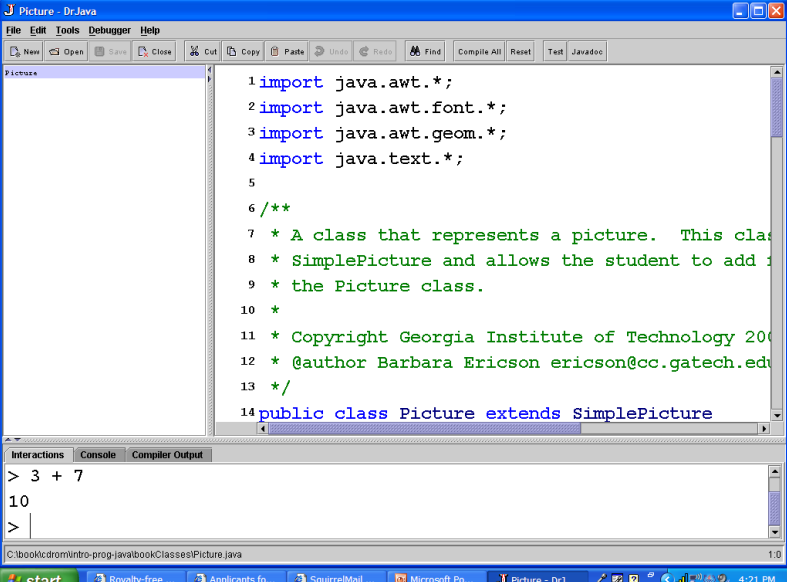
Math Operators in Java (+ * / - %)

- Addition: 3 + 4
- Multiplication: 3 * 4
- Division: 3 / 4
- Subtraction: 3 – 4
- Negation: - 4
- Modulo (Remainder): 10 % 2 and 11 % 2

Math Operators Exercise

- Open DrJava and do the following in the interactions pane

  - Subtract 7 from 9

  - Add 7 to 3

  - Divide 3 by 2

  - Divide 4.6 by 2

  - Multiply 5 by 10

  - Find the remainder when you divide 10 by 3

Why is the result of 3 / 2 = 1?

- Java is a **strongly typed language**
  - Each value has a type associated with it
  - Tells the computer how to interpret the number
    - It is an integer, floating point, letter, etc
- The compiler determines the type if it isn't specified (literals)
  - 3 is an integer
  - 3.0 is a floating point number (has a fractional part)
- The result of an operation is in the same type as the operands
  - 3 and 2 are integers so the answer is an integer 1

Casting

- There are other ways to solve the problem of 3 / 2 has a result of 1

- You can make one of the values floating point by adding .0

  - 3.0 / 2

  - 3 / 2.0

- The result type will then be floating point

- Or you can cast one of the values to the primitive types: float or double

  - (double) 3 / 2

  - 3 / (float) 2

Casting Exercise

- Use casting to get the values right for splitting up a bill for 3 people of 19 dollars.

- Try it first with a calculator

- Try it in DrJava without casting

- Try it in DrJava with casting

Java Primitive Types

- Integers (numbers without fractional parts) are represented by

  - The types: int or short or long
  - 235, -2, 33992093, etc

- Floating point numbers (numbers with fractional parts) are represented by

  - The types: double or float
  - 3.233038983 -423.9, etc

- A single character is represented by

  - The type: char
  - 'a' 'b' 'A' etc

- True and false values are represented by

  - The type: boolean
  - true or false

Why so Many Different Types?

- They take up different amounts of space

- They have different precisions

- Usually use int, double, and boolean

    - byte uses 8 bits (1 byte) 2's compliment
    - short uses 16 bits (2 bytes) 2's compliment
    - int uses 32 bits (4 bytes) 2's compliment
    - long uses 64 bits (8 bytes) 2's compliment
    - float uses 32 bits (4 bytes) IEEE 754
    - double uses 64 bits (8 bytes) IEEE 754
    - char uses 16 bits (2 bytes) Unicode format

Sizes of Primitive Types

byte | 8 bits

short | 8 bits | 8 bits

int | 8 bits | 8 bits | 8 bits | 8 bits

long | 8 bits | 8 bits | 8 bits | 8 bits | 8 bits | 8 bits | 8 bits | 8 bits

float | 8 bits | 8 bits | 8 bits | 8 bits

double | 8 bits | 8 bits | 8 bits | 8 bits | 8 bits | 8 bits | 8 bits

char | 8 bits | 8 bits

Types Exercise

- Which type(s) take up the most space?
- Which type(s) take up the least space?
- What type would you use for
  - The number of people in your family
  - A grade
  - The price of an item
  - The answer to do you have insurance
  - The number of people in the class
  - The number of people in your school
  - The number of people in your state

Floating Point Numbers

- Numbers with a fractional part
  - 6170.20389
- Stored as binary numbers in scientific notation -52.202 is -.52202 x 10^2
  - The sign (1 bit)
  - The digits in the number (mantissa)
  - The exponent (8 bits)
- Two types
  - float – 6-7 significant digits accuracy
  - double – 14-15 significant digits accuracy

Comparison (Relational) Operators

- Greater than >
  - 4 > 3 is true
  - 3 > 3 is false
  - 3 > 4 is false

- Less than <
  - 2 < 3 is true
  - 3 < 2 is false

- Equal ==
  - 3 == 3 is true
  - 3 == 4 is false

- Not equal !=
  - 3 != 4 is true
  - 3 != 3 is false

- Greater than or equal >=
  - 3 >= 4 is true
  - 3 >= 3 is true
  - 2 >= 4 is false

- Less than or equal <=
  - 2 <= 3 is true
  - 2 <= 2 is true
  - 4 <= 2 is false

Comparison Operators Exercise

- In DrJava
  - Try out the comparison operators in the interactions pane

    - with numbers
      3 < 4
      4 <= 4
      5 < 4
      6 == 6.0

    - with characters (single alphabet letter)
      Put single quote around a character
      'a' < 'b'
      'b' < 'a'
      'a' == 'a'

Operator Order

- The default evaluation order is
  - Negation -
  - Multiplication *
  - Division /
  - Modulo (remainder) %
  - Addition +
  - Subtraction -
- The default order can be changed
  - By using parenthesis
  - (3 + 4) * 2 versus 3 + 4 * 2

Math Operator Order Exercise

- Try 2 + 3 * 4 + 5

- Add parentheses to make it clear what is happening first

- How do you change it so that 2 + 3 happens first?

- How do you change it so that it multiplies the result of 2 + 3 and the result of 4 + 5?

Printing Output to the Console

- One of the things you often want to do in a program is output the value of something

- In Java the way to print to the console is to use

  - System.out.println();

    - Will print out the value of the thing in the parentheses and a new line

  - System.out.print();

    - To print just the thing in the parentheses without a new line

A Semicolon (;) ends a Statement

- Java programs are made up of statements
    - Like sentences in English
- Java statements end in a semicolon not a period
    - The period is used to send a message to an object
        - System.out.**println**()
    - Or access data in the object
        - System.**out**.println()
- DrJava's interaction pane prints the result of statements without a semicolon
    - but not the result of statements with a semicolon
        - Use System.out.println(); to force output

Console Output Exercise

- Use **System.out.println()** to print the results of expression to the console

  - System.out.println(3 * 28);
  - System.out.println(14 – 7);
  - System.out.println(10 / 2);
  - System.out.println(128 + 234);
  - System.out.println("Hi" + "There");
  - System.out.println("128 + 234");
- Try using **System.out.print()** instead
  - What is the difference?

Strings

- Java has a type called: String
- A string is an object that has a sequence of characters in Unicode
  - It can have no characters (the null string "")
  - It can have many characters
    - "This is one long string with spaces in it."
  - Everything in a string will be printed out as it was entered
    - Even math operations "128 + 234"
- Java knows how to add strings
  - It returns a new string with the characters of the second string after the characters of the first
    - With no added space

Methods

- Two Types
  - Object method
    - Sent as a message to an object
    - Implicitly passed the current object
  - Class method
    - Sent as a message to a class

Method Exercise

- In DrJava's interaction pane try these

  - Object methods

    - "HI".toLowerCase()
    - "This is a string".indexOf("is")
    - "   This is ".trim()

  - Class methods

    - Math.abs(13)
    - Math.abs(-13)
    - Math.min(3,4)
    - Character.getNumericValue('A')

Message Always Have Parenthesis

- You can tell that out.println() is sending a message
  - Because of the ()
- Messages always have ()
  - Even if there are no parameters (arguments)
- If you are sending data along with a message it goes inside the parentheses
  - Separated by commas
  - Math.min(3,4);

Common Errors

- Did you make any mistakes when you typed in the examples?

  - If you use the wrong case it won't work

    > math.abs(-3)

    Error: Undefined class 'math'

  - If you misspell something it won't work

    > Mat.abs(-3)

    Error: Undefined class 'Mat'

    > Math.ab(-3)

    Error: No 'ab' method in 'java.lang.Math'

- Use the up arrow key in DrJava to bring up the previous statement and fix it

"Hi" is a String Object

- The compiler turns literal strings into string objects

  - Objects of the String class

  - In package java.lang

- Object methods are invoked by sending a message

  - with the same name as the method

  - the same type, number, and order of input parameters

API Exercise

- The Classes defined as part of the Java language are documented in the API

  - http://java.sun.com/j2se/1.5.0/docs/api/

- Find the documentation for the following classes

  - String and Math

  - Find documentation for the methods used in the previous exercise

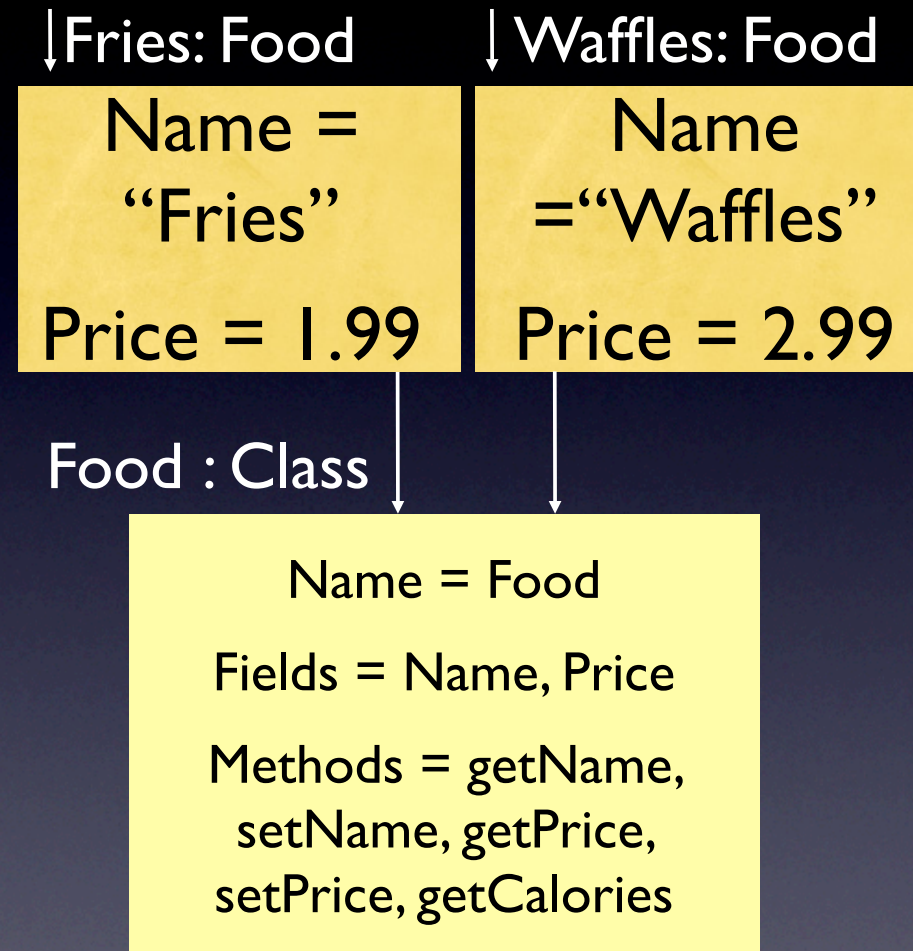 - Try out some other methods for these classes

Java Packages

- Java groups related classes into packages
- Common Packages
  - java.lang
    - Contains basic classes for the language
      - System, Math, Object, …
  - java.io
    - Contains classes for input and output
  - java.awt
    - Contains basic user interface classes
  - javax.swing
    - Contains more advanced user interface classes

Class Methods versus Object Methods

- In the API documentation how can you tell which are class methods and which are object methods?

  - Look for the keyword **static** on the method

  - If it has the keyword static then it is a **class** method

  - If there is no keyword static then it is an object method

What do Objects Look Like?

- Objects are created with space for their data

- Objects have a reference to the object that represents the class

  - Object of the class "Class"

↓Fries: Food

Name = "Fries"
Price = 1.99

↓Waffles: Food

Name ="Waffles"
Price = 2.99

Food : Class

Name = Food

Fields = Name, Price

Methods = getName, setName, getPrice, setPrice, getCalories

Java is Case Sensitive

- Some programming languages are case sensitive
  - Meaning that **double** isn't the same as **Double**
  - Or **string** isn't the same as **String**
- In Java primitive types are all lowercase
  - double, float, int,
- Class names start with an uppercase letter
  - So String and System are the names of classes

Java Naming Conventions

- In Java **only** Class names start with an uppercase letter
  - System, BufferedImage, Picture
- All other names start with lowercase letters but uppercase the first letter of each additional word
  - picture, fileName, thisIsALongName

Identifying Classes Exercise

- Which of these are primitive types, and which are the names of classes?

  - int
  - Picture
  - char
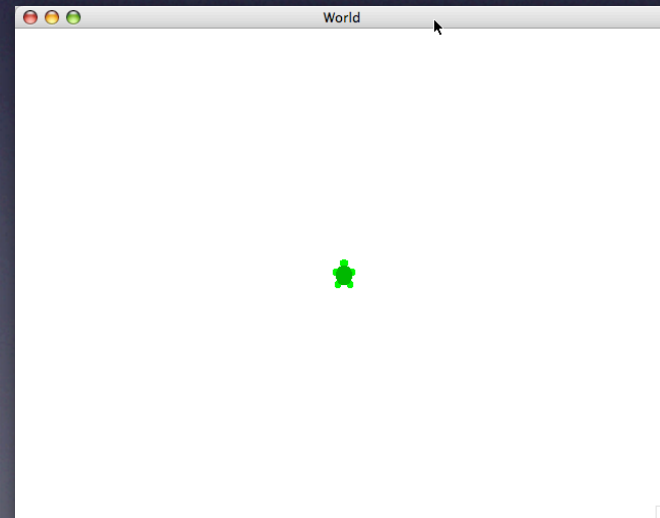 - Double
  - Math
  - double
  - Integer
  - String

Turtle Graphics

- Try the following in your codepad

```
World w = new World();

Turtle t = new Turtle(w);

t
```

Manipulating the turtle

- Right click on turtle1 and see the methods available

- forward(), backward(), clearpath(), getXPos(), getYPos(), moveTo(), penDown(), penUp(), turn(), turnLeft(), turnRight(), setColor(), setName()

- Can you make your turtle draw a square?

Turtle Graphics

- Create a new project, with a new class TurtleWorld

- Create a variable in the class of type World and name it w

- Create a variable in the class of type Turtle and call it t

- Create a constructor and in it create a World object and a Turtle Object

- Lets create some method in our TurtleWorld: square() - make it draw a square

Turtle Graphics

- What if you wanted to have many turtles?

- Lets create an array and put all turtles there.

- Lets create a method that adds turtles to the end of the array.

- Lets modify the square() method to take an index identifying the turtle

- Can you create a method that moves all turtles at once?

Coming attractions

- Monday
  - Quiz 11 due 10:00 am