

Intro to Java

- Anatomy of a Class & Terminology
- Running and Modifying a Program

CompSci 4

Intro to Java

2.1

Why know the lingo?

- ❖ It's difficult to read the textbooks if you don't understand the words
- ❖ Your compiler error messages use specific words with specific meanings
- ❖ You need to be able to express your questions so others can understand them
- ❖ The answers you receive will use the lingo

CompSci 4

Intro to Java

2.3

The Plan

- ❖ Go over MoveTest.java
 - ❑ Similar to Horstmann p. 48
- ❖ Basic coding conventions
- ❖ Review with GreeterTest.java (Horstmann)
- ❖ More terminology with Greeter.java (Horstmann)
- ❖ Homework 0 reminder
- ❖ Homework 1 assigned (due in 1 week)

CompSci 4

Intro to Java

2.2

Terminology to Know

- ❖ Package
- ❖ Class
- ❖ Import
- ❖ Keyword
- ❖ Public
- ❖ Object
- ❖ Identifier
- ❖ Declaration
- ❖ Definition
- ❖ Body
- ❖ Static
- ❖ Void
- ❖ Return
- ❖ Method
- ❖ Main
- ❖ Parameter
- ❖ String
- ❖ Array
- ❖ Type
- ❖ Variable
- ❖ Local
- ❖ Constructor
- ❖ Initialize
- ❖ Assign
- ❖ Arguments
- ❖ Comments
- ❖ Calling a method
- ❖ System.out.println

CompSci 4

Intro to Java

2.4

MoveTester.java

```
import java.awt.Rectangle;

public class MoveTest
{
    public static void main(String[] args)
    {
        Rectangle cerealBox = new Rectangle(5, 10, 20, 30);
        // move the rectangle
        cerealBox.translate(15, 25);
        // print the moved rectangle
        System.out.println(cerealBox);
    }
}

Prints
java.awt.Rectangle[x=20, y=35, width=20, height=30]
```

CompSci 4

Intro to Java

2.5

java package
MoveTest.java abstract windowing toolkit package
import java.awt.Rectangle; Rectangle class

import means to bring into the program classes or packages not in the package java.lang
import is a keyword – it is a word with a special meaning

Prints
java.awt.Rectangle[x=20, y=35, width=20, height=30]

CompSci 4

Intro to Java

2.6

public means usable by everything, public is also a keyword

MoveTest.java

```
import java.awt.Rectangle;
public class MoveTest
{
    public static void main(String[] args)
    {
        Rectangle cerealBox = new Rectangle(5, 10, 20, 30);
        // move the rectangle
        cerealBox.translate(15, 25);
        // print the moved rectangle
        System.out.println(cerealBox);
    }
}
```

Prints

java.awt.Rectangle[x=20, y=35, width=20, height=30]

class means instruction and data for making objects, class is a keyword

MoveTest is the name of the class
A class name must match the file name.
Names are also called identifiers.
Identifiers and keywords are mutually exclusive.

CompSci 4

Intro to Java

2.7

MoveTest.java

import java.awt.Rectangle;
class declaration

public class MoveTest
class definition.
{ } Starts and ends class body

CompSci 4

2.8

MoveTest.java

```
import java.awt.Rectangle;  
  
public class MoveTest  
{  
    public static void main(String[] args)  
    {  
        Rectangle cerealBox = new Rectangle(5, 10, 20, 30);  
        // move the rectangle  
        cerealBox.translate(15, 25);  
        // print the moved rectangle  
        System.out.println(cerealBox);  
    }  
}  
  
Prints  
java.awt.Rectangle[x=20, y=35, width=20, height=30]
```

CompSci 4

Static means one per class

void means no return value

main is the name
of the method

Intro to Java

2.9

MoveTest.java

```
import java.awt.Rectangle;  
  
public class MoveTest  
{  
    public static void main(String[] args)  
    {  
        Rectangle cerealBox = new Rectangle(5, 10, 20, 30);  
        // move the rectangle  
        cerealBox.translate(15, 25);  
        // print the moved rectangle  
        System.out.println(cerealBox);  
    }  
}  
  
Prints  
java.awt.Rectangle[x=20, y=35, width=20, height=30]
```

CompSci 4

String is a sequence of characters

[] means an array

args is a parameter

Intro to Java

2.10

MoveTest.java

```
import java.awt.Rectangle;  
  
public class MoveTest  
{  
    public static void main(String[] args)  
    {  
        Rectangle cerealBox = new Rectangle(5, 10, 20, 30);  
        // move the rectangle  
        cerealBox.translate(15, 25);  
        // print the moved rectangle  
        System.out.println(cerealBox);  
    }  
}  
  
Prints  
java.awt.Rectangle[x=20, y=35, width=20, height=30]
```

CompSci 4

method declaration

{ } Starts and ends
method body

method body

method definition

Intro to Java

2.11

MoveTest.java

```
import java.awt.Rectangle;  
  
public class MoveTest  
{  
    public static void main(String[] args)  
    {  
        Rectangle cerealBox = new Rectangle(5, 10, 20, 30);  
        // move the rectangle  
        cerealBox.translate(15, 25);  
        // print the moved rectangle  
        System.out.println(cerealBox);  
    }  
}  
  
Prints  
java.awt.Rectangle[x=20, y=35, width=20, height=30]
```

CompSci 4

Rectangle is a type (also a class)

cerealBox is a variable

Creates a Rectangle
Calls the constructor of
the Rectangle class

Intro to Java

2.12

MoveTest.java

```
import java.awt.Rectangle; Declaring the cerealBox variable of type Rectangle

public class MoveTest
{
    public static void main(String[] args)
    {
        Rectangle cerealBox = new Rectangle(5, 10, 20, 30); // move the rectangle
        cerealBox.translate(15, 25); // print the moved rectangle
        System.out.println(cerealBox);
    }
}

Prints
java.awt.Rectangle[x=20, y=35, width=20, height=30]
```

CompSci 4

Initializing the cerealBox variable

Intro to Java

2.13

MoveTest.java

```
import java.awt.Rectangle;

public class MoveTest
{
    public static void main(String[] args)
    {
        Rectangle cerealBox = new Rectangle(5, 10, 20, 30); // move the rectangle
        cerealBox.translate(15, 25); // print the moved rectangle
        System.out.println(cerealBox);
    }
}

Prints
java.awt.Rectangle[x=20, y=35, width=20, height=30]
```

CompSci 4

Assignment operator
Pronounced “gets”

1. Computes the right hand side
2. Assigns value to left hand side

Intro to Java

2.14

MoveTest.java

```
import java.awt.Rectangle; Arguments – order matters

public class MoveTest
{
    public static void main(String[] args)
    {
        Rectangle cerealBox = new Rectangle(5, 10, 20, 30); // move the rectangle
        cerealBox.translate(15, 25); // print the moved rectangle
        System.out.println(cerealBox);
    }
}

Prints
java.awt.Rectangle[x=20, y=35, width=20, height=30]
```

CompSci 4

Intro to Java

2.15

MoveTest.java

```
import java.awt.Rectangle;

public class MoveTest
{
    public static void main(String[] args)
    {
        Rectangle cerealBox = new Rectangle(5, 10, 20, 30); // move the rectangle
        cerealBox.translate(15, 25); // print the moved rectangle
        System.out.println(cerealBox);
    }
}

Prints
java.awt.Rectangle[x=20, y=35, width=20, height=30]
```

CompSci 4

Comments
Ignored by compiler

Intro to Java

2.16

MoveTest.java

```
import java.awt.Rectangle;  
  
public class MoveTest  
{  
    public static void main(String[] args)  
    {  
        Rectangle cerealBox = new Rectangle(5, 10, 20, 30);  
        // move the rectangle  
        cerealBox.translate(15, 25);  
        // print the moved rectangle  
        System.out.println(cerealBox);  
    }  
}  
  
Prints  
java.awt.Rectangle[x=20, y=35, width=20, height=30]
```

Local variables are declared in method bodies

Calling the translate method
On the cerealBox object

Rectangle object
Also a local variable

CompSci 4

Intro to Java

2.17

MoveTest.java

```
import java.awt.Rectangle;  
  
public class MoveTest  
{  
    public static void main(String[] args)  
    {  
        Rectangle cerealBox = new Rectangle(5, 10, 20, 30);  
        // move the rectangle  
        cerealBox.translate(15, 25);  
        // print the moved rectangle  
        System.out.println(cerealBox);  
    }  
}  
  
Prints  
java.awt.Rectangle[x=20, y=35, width=20, height=30]
```

Calling the println method
For console output

CompSci 4

Intro to Java

2.18

Why know and follow the Java Coding Conventions?

- ❖ Helps understand code
 - makes purpose of identifiers clear
 - delineates separate pieces of code
 - assists in avoiding syntax errors
- ❖ Expected if source code is to be viewed at any time by anyone other than the original author
- ❖ Helps standardize

CompSci 4

Intro to Java

2.19

Coding Conventions

- ❖ Capitalization
 - Class identifier
 - Variable identifier
 - Method identifier
- ❖ Indentation
 - Braces
 - Body of code (also called a code block)
- ❖ See course webpage for a complete description

CompSci 4

Intro to Java

2.20

GreeterTest.java

```
public class GreeterTest
{
    public static void main(String[] args)
    {
        Greeter worldGreeter = new Greeter("World");
        System.out.println(worldGreeter.sayHello());

        Greeter daveGreeter = new Greeter("Dave");
        System.out.println(daveGreeter.sayHello());
    }
}
```

CompSci 4

Intro to Java

2.21

Greeter.java

```
public class Greeter
{
    public Greeter(String aName)
    {
        name = aName;
    }

    public String sayHello()
    {
        String message = "Hello, " + name + "!";
        return message;
    }

    private String name;
}
```

Constructor
Used to initialize instance variables
Has no return type, not even void
Name is the same as the class name

Declaration of instance variable
Outside method body
Inside class body

CompSci 4

Intro to Java

2.22

Greeter.java

```
public class Greeter
{
    public Greeter(String aName)
    {
        name = aName;
    }

    public String sayHello()
    {
        String message = "Hello, " + name + "!";
        return message;
    }

    private String name;
}
```

Empty parameter list

String concatenation

Private means only
accessible within
class body

CompSci 4

Intro to Java

2.23

Introduction to Java Downloading Source Code

- ❖ Start up Eclipse
 - (In ICC: In applications folder on desktop)
- ❖ Set snarf site to:
<http://www.cs.duke.edu/courses/spring06/cps004/snarf>

CompSci 4

Intro to Java

2.24

Assignment #1

❖ **Create your Class Web Page**

□ See assignment on web

❖ **Due Tuesday, 1/24**

❖ **(Assignment #0 Due Today!)**