
CSE 142

Creating a Class

1/14/2002 (c) University of Washington, 2001-2 E-1

Overview

- **Quick Review**
 - Creating, naming, and using objects
 - Expressions and statements
 - Types and Errors
 - Messages and parameters
- **Today**
 - Build new kinds of objects
- **Reading**
 - Dugan notes: parts of Ch. 6
 - Niño & Hosch: Skim ch. 4-5 as needed

1/14/2002 (c) University of Washington, 2001-2 E-2

Classes and Objects

- **The Java World**
 - Objects interact with each other by sending and reacting to messages
 - Objects are instances of *classes*
 - A class defines a new type and is a blueprint for individual instances (objects) of that class
- **The fundamental programming task in Java is creating or modifying classes**

1/14/2002 (c) University of Washington, 2001-2 E-3

Our Own Shape: A House

- We can use rectangles, triangles, etc. to create a model of what a house looks like.
- Let's do it!

1/14/2002 (c) University of Washington, 2001-2 E-4

Wanted: A House Class

- We would like to package these statements so we can create House objects, just as we can create Rectangle objects now
 - We want to invent a new *abstraction*, i.e., a new word in our vocabulary of shapes

```
House home = new House();
```

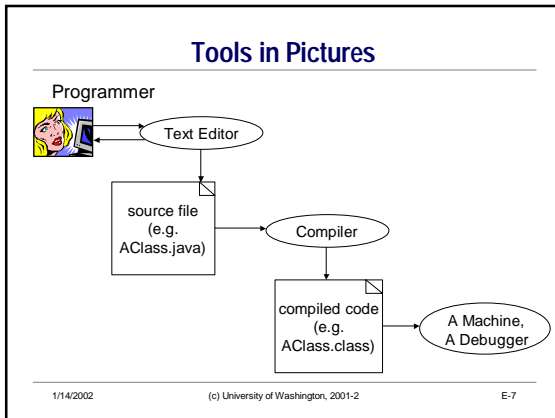
- We want to define our own *class*

1/14/2002 (c) University of Washington, 2001-2 E-5

Tools for Writing Classes

- Jeva doesn't allow us to define new classes. \perp
- BlueJ does! J
- BlueJ includes several tools:
 - a *text editor* to write a Java source (text) file containing our class definition
 - a *compiler* to translate Java source into a form that can be run, e.g. by Jeva
 - a graphical tool to create new objects and send them messages
 - a *debugger* to help us watch our programs run

1/14/2002 (c) University of Washington, 2001-2 E-6



Developing Software with a Compiler

- It usually involves (at least) the following steps:
 - 1: Edit your source code
 - 2: Compile your source code
 - 3: Possibly repeat steps 1 and 2. (Why?)
 - 4: Run the program
 - 5: Possibly go back to step 1 (Why?)
- Notice the differences compared to the interpreter:
 - "Batch mode" (compiler) vs. "interactive mode" (interpreter)

1/14/2002 (c) University of Washington, 2001-2 E-8

Defining a Class

- In a file named `House.java`, write and compile:

```
/* A house object */
public class House {

}
```
- Then we can create instances of the `House` class
 - In Jeva (or in Java code) we'd type
`House home = new House();`
 - In BlueJ, right-click on the class icon (`House`) and create a new instance (underneath it's the same)
Can inspect new object - doesn't contain much (yet!)

1/14/2002 (c) University of Washington, 2001-2 E-9