Ch3: Classes

24 January 2007
CMPT167
Dr. Sean Ho
Trinity Western University

TRINITY WESTERN UNIVERSITY

What's on for today

- Operator precedence
- Java primitive types
- public/private keywords
- set/get methods
- Subclasses, instances, attributes
- Interfaces
- More on writing and running applets

Review: operator precedence

- In order from most tightly bound first:
  - Parentheses: ()
  - Unary postfix (r to l): x++, x--
  - Unary prefix (r to l): ++x, --x, +x, -x, (type) x
  - Multiplicative: *, /, %
  - Additive: +, -
  - Relational: <, >, <=, >=
  - Equality: ==, !=,
  - Conditional (r to l): ?:
  - Assignment (r to l): =, +=, -=, *=, /=, %=, etc.

Java primitive types

- boolean (1 byte): true, false
- char (2 bytes): Unicode, '\u0000' to '\uFFFF'
- byte (1 byte): -128 to +127
- short (2 bytes): -32768 to +32767
- int (4 bytes): -2^{31} to $+2^{31}-1$
- long (8 bytes): -2^{65} to $+2^{65}-1$
- float (4 bytes): +/- 1.40129846432481707e-45 to 3.40282346638528860e+38
- double (8 bytes): +/- 4.94065645841246544e-324 to 1.79769313486231570e+308

public/private keywords

- So far most of our classes/attributes/methods have been declared public
- The private keyword specifies that only methods within this class can access this entity:

```
class Student {
    private String name;
}
Student s1 = Student();
s1.name;      // error!
```

- This is for information hiding: prevent others from directly accessing/modifying an entity.

Set/get methods

- A common idiom is to declare instance variables private but provide public set/get methods:

```
class Student {
    private String name;
    public String getName() { return name; }
    public setName(String n) { name = n; }
}
```

- Advantages of set/get over just declaring public?
  - Control access to the instance variable
    - Can add error checking
  - Hides underlying storage type of variable
    - Can upgrade to different data structure later

Subclasses, instances, attributes

- Recall classes are user-defined container types
- A subclass inherits attributes and methods from the superclass
- Subclasses should be seen as specializations of the superclass: "A is a kind of B"
- Instances should be seen as examples of a class: "A is a B"
- Attributes should be seen as components or parts of a class: "A has a B"

Example

- class Mammal { Heart h; }
- class Dog extends Mammal { void bark(); }
- class Cat extends Mammal { void meow(); }
- Dog fido = new Dog();
- Cat smokey = new Cat();

- "A Dog is a kind of Mammal."
- "fido is a Dog."
- "fido is a Mammal."
- "fido has a Heart."
- "smokey can meow()."

Interfaces

- An interface is a set of methods that a class implements
  - public interface Speaker { public void speak(); }
  - class Dog extends Mammal implements Speaker {
    - void bark() { System.out.println("Woof!"); }
    - public void speak() { bark(); }
  - }
  - class Cat extends Mammal implements Speaker {
    - void meow() { System.out.println("Meow!"); }
    - public void speak() { meow(); }
  - }
- Compare fido.speak() with smokey.speak()

More on applets

- Addition example applet ("Lab0")
  - import java.applet.Applet; // the Applet class
  - import java.awt.*; // abstract window toolkit
  - public class Addition extends Applet implements ActionListener {
- Only one public class per file
  - Named same as the file
- Extends: subclass inherits from Applet
- Implements: ActionListener interface
  - Must implement the following method:
  - public void actionPerformed(ActionEvent e)

Running an applet

- Compile an applet using javac as usual
- Run: in Eclipse: it will pop up a window
- Run: in a web page:
  - Write a small HTML file embedding the applet:
    - <object> (IE), or <applet>, or both
    - See Addition.html and TicTacToe.html
    - See Sun's recommendations
- Run: using appletviewer:
  - appletviewer Addition.html
    - HTML file, not the applet .class file directly

Review of today

- Operator precedence
- Java primitive types
- public/private keywords
- set/get methods
- Subclasses, instances, attributes
- Interfaces
- More on writing and running applets

TODO

- Lab1a due tonight:
  - Selection structure
  - Swing program: see "SayHello" example, or
  - Java Applet: see "Lab0" (Addition) template
- Lab1b due next Wed 31Jan:
  - Repetition structure