1.00 Lecture 7

Java Classes and Objects

Reading for next time: Big Java: sections 8.1-8.5

Classes

- **A class is a pattern or template from which objects are made**
  - **You may have many birds in a simulation**
    - **One bird class (or more if there's more than one type of bird)**
    - **Many bird objects (actual instances of birds)**
- **Objects are instances of classes**
  - **Class: Student Object: Joe Smith Object: Jane Wang**
  - **Class: Building Object: Building 10 Object: Building 7**
  - **Class: Street Object: Mass Ave Object: Vassar St**

Class Definition

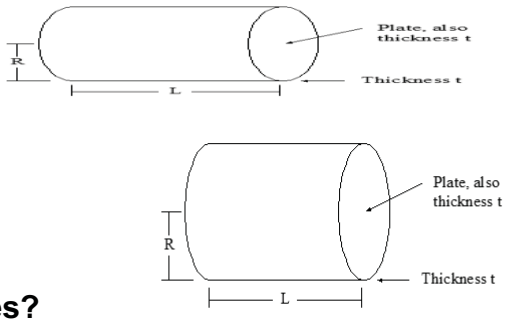
- **Classes contain:**
  - **Data (members, fields)**
    - **Simple data types, like int or double (e.g. bird weight)**
    - **Objects (e.g. bird beak)**
  - **Methods (functions, procedures)**
    - **Behaviors that an object can execute (e.g. bird flies/moves)**
    - **Special method: constructor (brings object into existence)**
- **Classes come from:**
  - **Java class libraries: JOptionPane, System, Math, etc. There are several thousand classes (see Javadoc)**
  - **Class libraries from other sources: Web, fellow students…**
  - **Classes that you write yourself**
- **Classes are usually the nouns in a problem statement (e.g. bird)**
  - **Data members are also nouns (e.g., weight)**
- **Methods are usually the verbs (e.g. flies)**

Building Classes

- **Classes hide their implementation details from the user (programmer using the already-written class):**
  - **Their data is not accessed directly, and the details are not known to 'outside' objects or programs.**
  - **Data is almost always `private` (keyword).**
- **Objects are used by calling their methods/behaviors.**
  - **The outside user knows what methods the object has, and what results they return.**
  - **The details of how their methods are written are not known to 'outsiders'**
  - **Methods are usually `public` (keyword).**
    - **By insulating the rest of the program from each object's details, it is much easier to build large programs correctly, and to reuse objects from previous work.**
    - **This is called encapsulation or information hiding.**
- **Access: public, private, (package, protected)**

Tank Class

Two tanks, each with:	t0	t1
- Radius R	0.5m	1.0m
- Length L	4.0m	1.0m
- Thickness t	0.04m	0.04m



Tank dimensions

- How many classes?
- How many objects?

Tank Class

```
public class Tank {
   // Data members–private. These are outside any method.
   private double radius;
   private double length;
   private double thickness;
   // Their scope is until ending } of class

   // Constructor or existence method–public (no return type ever)
   public Tank(double r, double len, double t) {
        radius = r;
        length = len;
        thickness = t;
   }
   // Methods on the next slide. There is no main() in this class.
```

Tank Class

```
// 'Get' methods–respond to messages from other objects--public

// Cylinder wall mass= Lπ((R+t)² – R²)
public double getMassCylinderWalls() {
     double mass= length*Math.PI*
      ((radius + thickness)*(radius + thickness)- radius*radius);
     return mass;
}

// Cylinder end mass= 2πt(R+t)²
public double getMassEnds() {
     double mass= 2.0*Math.PI*thickness*
       (radius + thickness)*(radius + thickness);
     return mass;
}

public double getMass() {          // We ignore density
     return getMassCylinderWalls() + getMassEnds();
}
```

TankTest Class

```
public class TankTest {
   public static void main(String[] args) {
        // Create two Tank objects
        Tank t0= new Tank(0.5, 4.0, 0.04);
        Tank t1= new Tank(1.0, 1.0, 0.04);

        // Ask object t0 to tell you its mass
        double t0wall= t0.getMassCylinderWalls();
        double t0end= t0.getMassEnds();
        double t0mass= t0.getMass();
        System.out.println("t0: "+ t0wall + " " + t0end +
                " " + t0mass);

        // Ask object t1 to tell you its mass
        double t1wall= t1.getMassCylinderWalls();
        double t1end= t1.getMassEnds();
        double t1mass= t1.getMass();
        System.out.println("t1: "+ t1wall + " " + t1end +
               " " + t1mass);
   }
}
```

Exercise, part 1

- Download Tank and TankTest
- In class Tank, implement additional methods:
  - Compute volume V
    - $V = \pi R^2 L$
  - Compute weld length W
    - $W = 2\,[2\pi(R+t) + 2\pi R] = 8\pi(R + t/2)$
  - Compute tank cost C, using arguments c_1, c_2
    - $C = c_1 M + c_2 W$ (tank mass M)
- Save/compile
  - You'll write a main() next that uses these methods

Exercise, part 2

- In class TankTest
  - Call your methods to compute volume, weld length and cost on tanks t0 and t1
    - Cost of weld= \$30/m
    - Cost of steel (mass)= \$48,000/m^3
  - Print out volume, weld length and cost for t0, t1
- Save/compile and read with the debugger to make sure it's working correctly

Exercise, part 3

- In class Tank:
  - Write 3 additional methods:
    - public void setRadius(double r) {radius= r;}
    - And similar for length and thickness
- In class TankTest
  - After all the existing code:
  - Change tank t1to have a radius of 0.8, and a length of 1.5625
  - Leave thickness the same
  - Output the revised volume, mass, weld length and cost
  - (The new tank is almost the minimum cost tank for this volume.)
- Save, compile and read it with the debugger

Review questions

- Circle all true statements
- A class is:
  - A group of methods
  - A pattern to make objects
  - The same thing as an object
  - A program
- An object is:
  - A group of methods
  - The same thing as a class
  - A specific thing that follows a pattern defined by a class
  - A built-in data type, like int or double

More review questions

- **A constructor is:**
  - **The method called when an object is created**
  - **A special data member defined at startup**
  - **A method that can be invoked on an object**
  - **A method whose name is the same as the class and has no return value**
- **A data member is:**
  - **A variable that describes an object, or thing, made using the pattern of the class**
  - **A variable that is defined in a method of the class**
  - **A variable that can be seen by (is in scope for) all methods of the class**
  - **A variable that cannot be used by any method**

Still more questions

- **The new operator:**
  - **Creates a new object and calls its constructor**
  - **Creates a new class**
  - **Is used to declare a new built-in data type**
  - **Is optional and may be used to create a class**
- **Objects:**
  - **Cannot be data members in other objects**
  - **Cannot be passed as arguments to methods**
  - **Can be the return value of a method**
  - **Exist only in pre-written Java libraries and cannot be created by you**

1.00 / 1.001 / 1.002 Introduction to Computers and Engineering Problem Solving
Spring 2012