public class Student {
  ...

  /* Private instance variables */
  private String name;
  private int ID;
  private double unitsEarned;
}

- **private** so cannot be directly accessed outside class
- Instance variables keep values between method calls
  - **Scope** of instance variables is lifetime of the object
- Stores “state” of the object
public class Student {

    /* Sets name and id for student */
    public Student(String studentName, int studentID) {
        name = studentName;
        ID = studentID;
    }

    ...

}

• **Name of class** is used as constructor name
• **Constructor does not specify return type**
• **Responsible for initializing object**
• **It is called when an object is created**
Implementing Constructor

/* Sets name and id for student */
public Student(String studentName,
               int studentID) {
    name = studentName;
    ID = studentID;
}

Implementing Constructor

/* Sets name and id for student */
public Student(String name,
               int ID) {
    name = name;
    ID = ID;
}

Um, there’s a problem here...

Parameters name and ID are just getting assigned to themselves!
Implementing Constructor

/* Sets name and id for student */
public Student(String name,
               int ID) {
    this.name = name;
    this.ID = ID;
}

• **this** refers to *receiver object*
• Automatically available to methods of a class
Implementing Constructor

/* Sets name and id for student */
public Student(String studentName,
               int studentID) {
    name = studentName;
    ID = studentID;
}

• Just give parameters distinct names from member variables
• Preferred route (less thinking…)}