Lecture 7
Objects and Classes
An Introduction to Data Abstraction
MIT AITI
June 13th, 2005
What do we know so far?

• Primitives: int, double, boolean, String*
• Variables: Stores values of one type.
• Arrays: Store many of the same type.
• Control Structures: If-then, For Loops.
• Methods: Block of code that we can pass arguments to and run anytime.
• Is this all we need?
So what’s the problem?

- Some data “sticks” together.
  - String[] names
  - int[][] grades
- Methods start to get complicated.
- Methods can only return one type.
- Programmers don’t want to think about all the underlying types.
Abstraction

• Objects are tools for abstraction.
• We abstract away details to deal with complex problems.
• Abstraction is a fundamental concept in computer science.
• There can be too much abstraction.
• The art is knowing which details to hide away and which to preserve.
What is an object?

• Objects have two parts:
  - State: Properties of an object.
  - Behavior: Things the object can do.

• Car Example:
  - State: Color, engine size, automatic
  - Behavior: Brake, accelerate, shift gear

• Person Example:
  - State: Height, weight, gender, age
  - Behavior: Eat, sleep, exercise, study
What is an Object?

Figures removed for copyright reasons.

See [http://java.sun.com/docs/books/tutorial/java/concepts/object.html](http://java.sun.com/docs/books/tutorial/java/concepts/object.html)

A Generic Object  An Bicycle Object
Why use objects?

• Modularity: Once we define an object, we can reuse it for other applications.

• Information Hiding: Programmers don’t need to know exactly how the object works. Just the interface.

• Example:
  – Different cars can use the same parts.
  – You don’t need to know how an engine works in order to drive a car.
Our first Class: LightSwitch

• `class LightSwitch {`
  
  `boolean on = true;`

  `}`

• The keyword `class` tells java that we’re defining a new type of Object.
• Classes are a blueprint.
• Objects are instances of classes.
• Everything in Java (except primitives) are Objects and have a Class.
Classes

A Bicycle Class

Two instances of the Bicycle Class

Figures removed for copyright reasons.

See "MyBike" and "YourBike" figures at http://java.sun.com/docs/books/tutorial/java/concepts/class.html
Our first Class: LightSwitch

• `class LightSwitch {
  boolean isOn = true;
}

• What state do LightSwitches have?
• State stored in fields; here it’s “isOn”.
• Fields are accessed using:
  – `variableName.fieldName`
  – (We’ll discuss other types of fields later.)
• What behavior do LightSwitches have?
Adding Behavior

• `class LightSwitch {`  
  `boolean isOn = true;`  
  `void flip() {`  
    `this.isOn = !this.isOn;`  
  `}`
`}`

• The `this` keyword means this particular object. Objects know themselves.

• `this.isOn` accesses the `isOn` field.

• What behavior does LightSwitch have now?
Using Objects

```java
public static void main(String[] args) {
    LightSwitch s = new LightSwitch();
    System.out.println(s.isOn);
    s.flip();
    System.out.println(s.isOn);
}
```

- The `new` keyword creates a new object.
- `new` must be followed by a constructor.
- We call methods like:
  - `variableName.methodName(arguments)`
- What does this code output?
Constructors

• Constructors provide objects with the data they need to initialize themselves, like “How to Assemble” instructions.
• Objects have a default constructor that takes no arguments, like LightSwitch().
• We can define our own constructors that take any number of arguments.
• Constructors have NO return type and must be named the same as the class:
  – ClassName(argument signature) { body }
Constructors

- `class LightSwitch {
    boolean isOn;
    void flip() {
        this.isOn = !this.isOn;
    }
    LightSwitch(boolean startState) {
        this.isOn = startState;
    }
}

- The `LightSwitch()` constructor no longer works. How do we instantiate an object?
Multiple Constructors

• We can have multiple constructors.
• Constructors can call each other.
  - LightSwitch() {
    LightSwitch(true);
  }
  - LightSwitch(boolean startState) {
    this.isOn = isOn;
  }
Pop Quiz

• What two properties do objects have?
• What is the difference between a class and an object?
• What is a field?
• What does the \textit{this} keyword mean?
• What does the \textit{new} keyword do?
• What is a constructor?
BankAccount Example

class BankAccount {
    double balance;
    String name;
    BankAccount(String name,
                double openBalance) {
        this.name = name;
        this.balance = openBalance;
    } // Continued next slide
    ...
}
BankAccount Example

...  
  double deposit(double amount) {
    balance += amount;
    return balance;
  }

  boolean withdraw(double amount) {
    if (amount < balance) {
      balance -= amount;
      return true;
    } else return false;
  }

} // End BankAccount Class
MIT OpenCourseWare
http://ocw.mit.edu

EC.S01 Internet Technology in Local and Global Communities
Spring 2005-Summer 2005

For information about citing these materials or our Terms of Use, visit: http://ocw.mit.edu/terms.