

Supplementary Slides on Objects and Classes

CS1: Java Programming
Colorado State University

Original slides by Daniel Liang
Modified slides by Chris Wilcox



cs163/cs164: Extra Slides on Classes and Objects

1

Classes

- ◆ Classes are the basis of object-oriented (OO) programming.
- ◆ They *encapsulate* functionality to form powerful *abstractions* of real world objects.
- ◆ What can classes be used for? Classes have many different uses, for example:

- Data Structures
- Code Libraries
- Java Programs
- Complex Objects



cs163/cs164: Extra Slides on Classes and Objects

2

Classes as Data Structures

- ◆ Just like a *struct* in C and C++ (no code), for example:

```
public class Student {  
    public String firstName;  
    public String lastName;  
    public Date birthDate;  
    public Address homeAddress;  
    public double gradePointAverage;  
}
```



cs163/cs164: Extra Slides on Classes and Objects

3

Classes as Code Libraries

- ◆ Just like a *library* in a procedural language (no data) like C or C++, for example:

```
public class Math {  
    public static final double PI = 3.14159;  
    public static double sin(double a) {...}  
    public static double exp(double a) {...}  
    public static double log(double a) {...}  
    public static double sqrt(double a) {...}  
}
```



cs163/cs164: Extra Slides on Classes and Objects

4

Classes as Small Programs

- ◆ Just like a *program* in a procedural language like C or C++, for example:

```
public class MySmallProgram{  
    public static void main(String args[]) {  
        System.out.println("Hello, World!");  
    }  
}
```



Classes as Large Programs

- ◆ Just like a *program* in a procedural language like C or C++, for example:

```
public class MyLargeProgram{  
    // lots of data  
    public static void main(String args[]) {  
        // lots of code  
    }  
    // lots of methods  
}
```



Classes as Complex Objects

- ◆ No comparable example in a procedural language like C or Pascal!

```
public class MyClass {  
    // class variables (static)  
    // instance variables (non-static)  
    // no main method  
    // class methods (static)  
    // instance methods (non-static)  
}
```



Using Different Class Types

```
// Data Structure  
Student students[] = new Student[100];  
students[0].firstName = "Christopher";  
// Code Library  
System.out.println(Math.sin(1.0));  
// Java Programs  
$ java MySmallProgram  
// Complex Objects  
MyClass myClass = new MyClass();  
myClass.initialize();
```

