

CS601: Principles of Software Development

Java Basics: Intro to Classes and Objects

Olga A. Karpenko

Announcements

- Look at the Syllabus and lecture notes
- Accept invitation to join Piazza
- Create a github account (use USF email)
- CS tutoring center will be open next week:
<http://tutoringcenter.cs.usfca.edu/>
- How to check if you have CS account:
[http://tutoringcenter.cs.usfca.edu/
resources/about-cs-accounts.html](http://tutoringcenter.cs.usfca.edu/resources/about-cs-accounts.html)

CS Background Survey

- Go to the course website -> Quizzes
- Access Code: JAVA
- Anonymous
- Will help me tailor this class to your needs

Java Basics

File Types

- Source code is stored in .java files
- Compiled byte-code is stored in .class files
- See HelloWorld.java
- Multiple files can be bundled and compressed into a jar file

How to Run a Java Program

- Compile
 - Compiler turns code into byte code
 - Ex: `javac HelloWorld.java`
- Run
 - JVM interprets the byte code
 - Ex: `java HelloWorld`

The main method of `HelloWorld` will be executed

How to Run from *Any Folder*

```
java -classpath /users/okar/code HelloWorld
```

OR

```
export CLASSPATH=$CLASSPATH:/users/okar/code
```

```
java HelloWorld
```

*In this example we assume HelloWorld.class is in /users/okar/code

Jar File

- JAR stands for Java Archive
- Archive file
- Used to aggregate several files into one file
- Ability to download an entire application in a single request

Jar File

- Compress:

```
jar cvf jarFile inputFile(s)
```

c option - create a JAR file

f option - output to a file

- Execute

```
java ClassName jarFile
```

Object-Oriented Programming

- Java is an object-oriented programming language
- Objects often represent real-world entities
- Examples:
 - An object representing a particular employee in a company
 - Each employee object manages data related to that employee

Objects

- An object has:
 - *state* - descriptive characteristics
 - *behaviors* - what it can do or what can be done to it

Ex: Bank Account Object

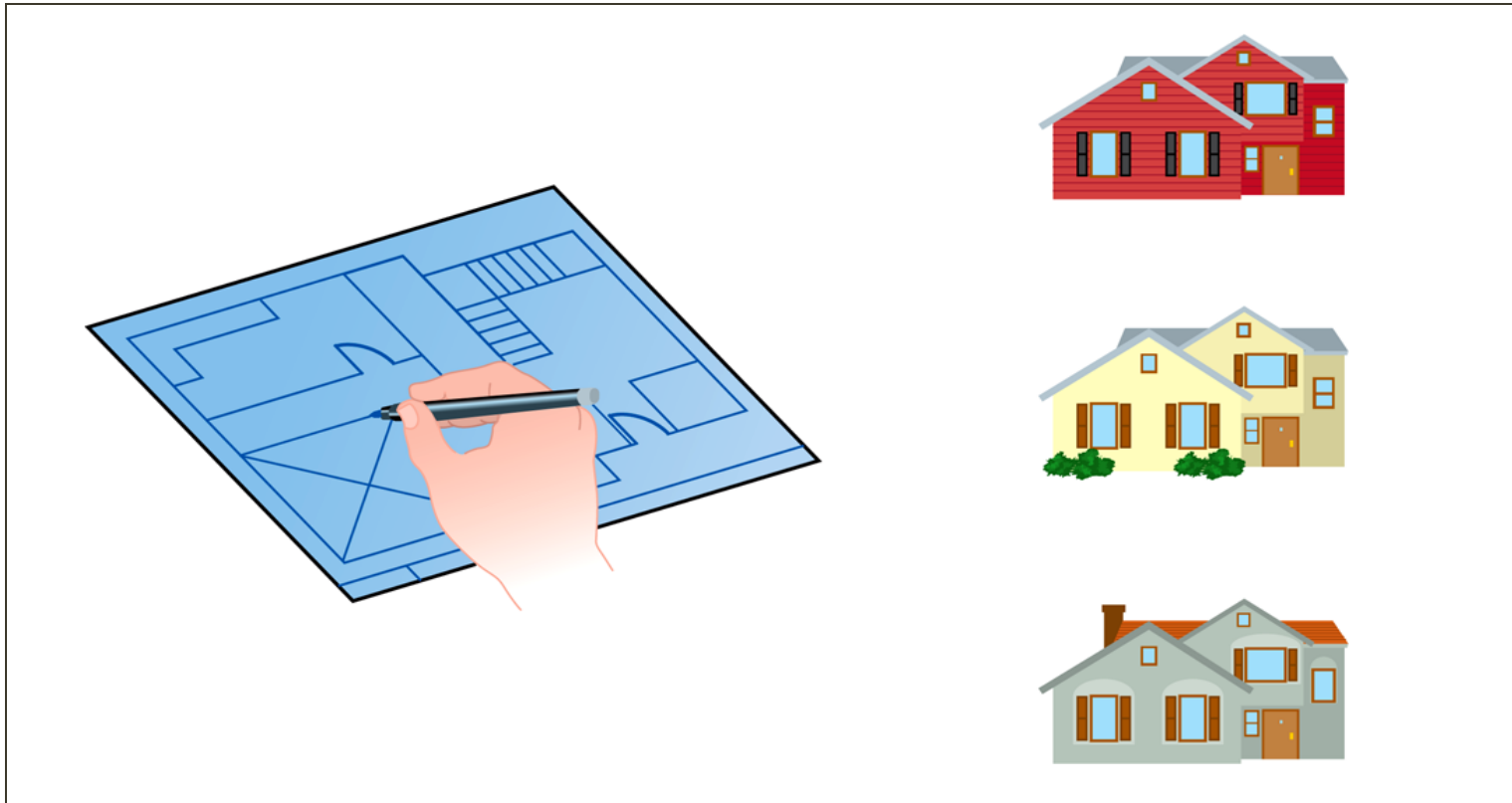
- The state: account number, owner's name and current balance
- The behaviors: deposit and withdraw

Classes

- An object is defined by a *class*
- A class is the *blueprint* of an object
- An object is an instance of the class
- Multiple objects can be created from the same class

Class = Blueprint

- One blueprint to create several similar, but different, houses:



Objects and Classes

A class
(the concept)



Objects:
(instances of this class)

John's Bank Account
Balance: \$5,257

Bill's Bank Account
Balance: \$1,245,069

Mary's Bank Account
Balance: \$16,833

Creating Objects

- A class name can be used as a type to declare an *object reference variable*

```
BankAccount account;
```

- No object is created with this declaration
- Object reference variable `account` will hold the *address* of an object
- The object itself must be created separately

Creating Objects

- We use the `new` operator to create an object
- Creating an object is called *instantiation*
- An object is an *instance* of a particular class

```
account = new BankAccount("Smith", 103952);
```



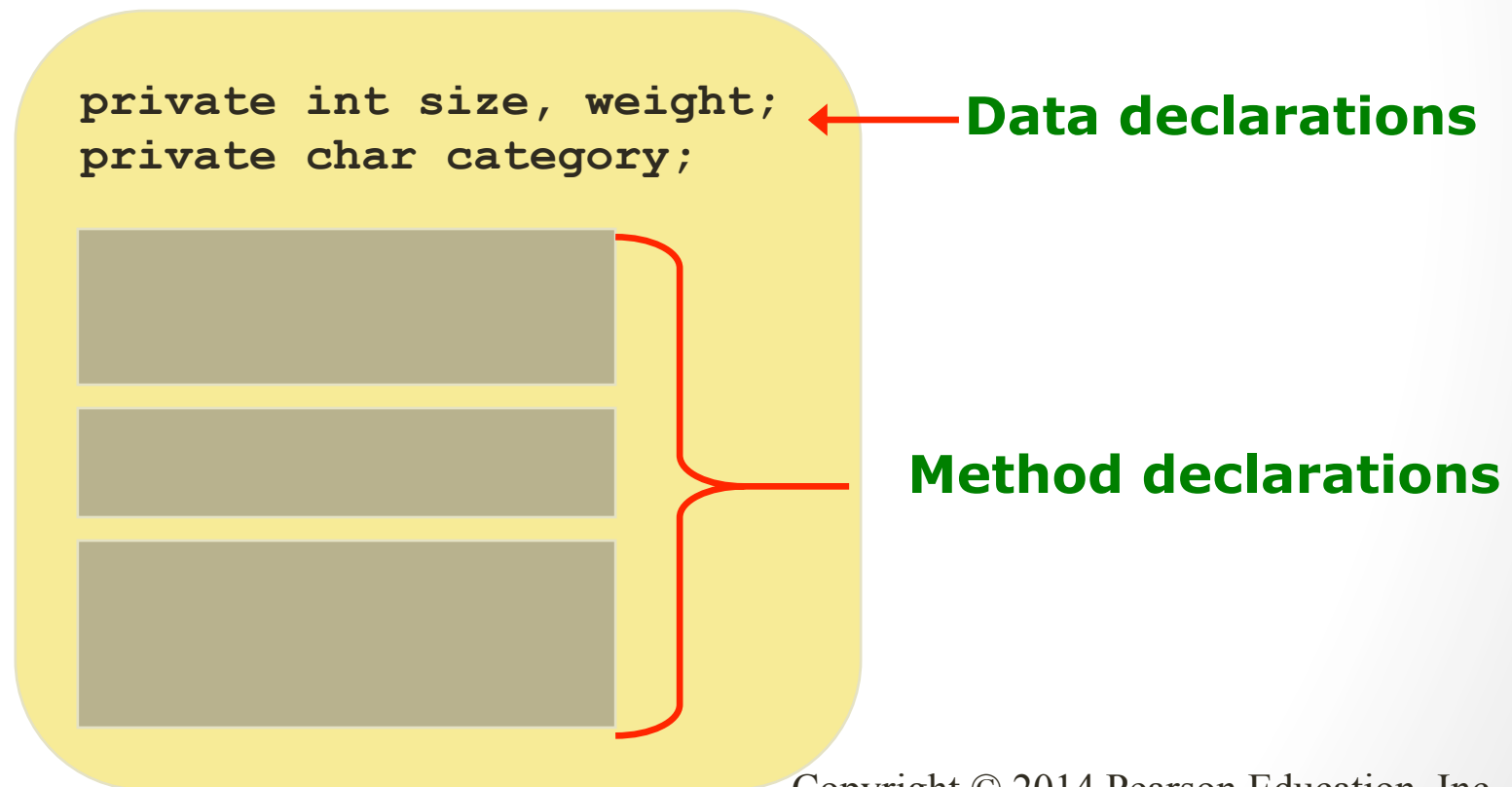
This calls the `BankAccount`'s *constructor*

Invoking Methods

- Use the *dot operator* to invoke object's methods
`double balance = account.getBalance();`
- Think of it as asking an object to perform a service

Writing Classes

- A class can contain data declarations and method declarations



Class Representing a Die

- Consider a six-sided die
- State: which face is showing
- Behavior: can be rolled
- Represent by class Die
- See `Die.java` and `DieExample.java`



Constructors

- Special methods that are invoked to construct objects
- Have the same name as the class
- Do not have a return type
- Are invoked using the new operator when an object is created
- A class can have multiple constructors with different parameters

The toString Method

- Returns a string that represents the object in some way
- It is called automatically when we specify the object in the println method
- It's good practice to define a toString method for each class
- Useful for testing and debugging