Java OO

CSC207 Winter 2015

Computer Science
UNIVERSITY OF TORONTO

Instance Variables

```
public class Circle {
    private String radius;
}
```

`radius` is an instance variable. Each object/instance of the class `Circle` has its own `radius` variable.

Constructors

A constructor has:

- the same name as the class
- no return type (not even `void`)

A class can have multiple constructors, as long as their signatures are different.

If you do not define a constructor, the compiler supplies one with no parameters and no body.

Once you define a constructor for a class, the compiler will no longer supply the default constructor.

`this`

`this` is an instance variable that you get without declaring it.

It is like `self` in Python.

Its value is the address of the object whose method has been called.

Defining Methods

- A method must have a return type declared. Use `void` if nothing is returned.
- The form of a return statement:

  ```
  return expression;
  ```

  If the expression is omitted or if the end of the method is reached without executing a return statement, nothing is returned.
- Must specify the accessibility. For now:

  `public` - callable from anywhere
  `private` - callable only from this class
- Variables declared in a method are local to that method.

Parameters

When passing an argument to a function, you pass what is in the variable's box:

- For class types, you are passing a reference. (Like Python.)
- For primitive types, you are passing a value.

This has important implications: aliasing!

Encapsulation

Convention: make all non-final instance variables *private*, which makes them accessible only within the class, or *protected*, which makes them accessible only within the package. When desired, give outside access using getter and setter methods.

Why make instance variables private?

We aim to hide implementation details and allow access to variables via a well-defined interface (the public methods for that class).

Access Modifiers

Classes can be declared public or package-private. Members of classes can be declared public, private, protected, or package-private.

Modifier	Class	Package	Subclass	World
`public`	Y	Y	Y	Y
`protected`	Y	Y	Y	N
no modifier (package-private)	Y	Y	N	N
`private`	Y	N	N	N