

Defining Classes in Java

CSC207 Fall 2016



Computer Science
UNIVERSITY OF TORONTO

Instance Variables

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

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

meaning different # of param

If you define no constructors, the compiler supplies one with no parameters and no body.

If you define any 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's 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
 - `public` – usable from anywhere
 - `private` – usable only from this class
 - `none` – usable from the package
 - `protected` – usable from the package and subclasses
where the subclass belong to a different package
- Variables declared in a method are local to that method.

Parameters

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

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

class type = object type = reference type

- For primitive types, you are passing a value.
(Python can't do anything like this.)

This has important implications!

You must be aware of whether you are passing a primitive or object.

Both class and primitive types are implemented as locations in memory. For a variable of a primitive type, the value of the variable is stored in the memory location assigned to the variable. However, a variable of a class type only stores the memory address of where the object is located (called reference) – not the values inside the object

Instance Variables and Accessibility

If an instance variable is private, how can client code use it?

Why not make everything public? So much easier!

Encapsulation

Think of your class as providing an abstraction, or a service.

- We provide access to information through a well-defined interface: the public methods of the class.
- We hide the implementation details.

What is the advantage of this “encapsulation”?

Java conventions

Make all non-final instance variables either:

- private: accessible only within the class, or
- no modifier: accessible only within the package.
 - This is also called "package private".
- protected: package plus subclasses in other packages.

When desired, give outside access using "getter" and "setter" methods.

Access Modifiers

Classes can be declared public or package-private.

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

Modifier	Class	Package	Subclass	World
public	Yes	Yes	Yes	Yes
protected	Yes	Yes	Yes	No
default (package private)	Yes	Yes	No	No
private	Yes	No	No	No