



002 Classes and Objects 2

Access control
Initializer blocks
enum types
Garbage collection

Variable Scope

The **scope** of a variable is the section of code from within which it can be accessed.

- The scope of local variables and parameters is limited to the containing method or block.
 - Local variables cease to exist when execution leaves the method or block.
- The scope of class and instance fields depends on the access control modifiers (`private`, `public`, etc).

Access Control

Access modifiers determine which other classes can access fields and methods:

- Top-level: **public** or package-private (no modifier).
- Member level: **public**, **protected**, package-private, or **private**

Modifier	Class	Package	Subclass	World
public	✓	✓	✓	✓
protected	✓	✓	✓	✗
<i>no modifier</i>	✓	✓	✗	✗
private	✓	✗	✗	✗

Class Members

The `static` modifier keyword identifies class variables and methods.

- A **class variable** is shared by all instances of the class.
- A **class method** is called without reference to an object
 - Cannot use `this` in a class method (there is no “this”).
 - A class method can only reference class fields.
 - Class methods can be referenced (called) from outside the class using the class name.

Initializer Blocks

Fields may be initialized when they are declared. They can also be initialized by **initializer blocks**, which can initialize fields using arbitrarily complex code (error handling, loops, etc.).

- A **static initializer** block consists of code enclosed by braces ‘{}’ and preceded by the `static` keyword. It runs when the class is first accessed.
- A **instance initializer** block does not have the `static` keyword, and runs before the constructor body of the class.

Enum Types

An **enumerated type** is defined with the `enum` keyword.

A variable of enum type must be one of a set of predefined values.

This is useful for defining non-numerical sets such as NORTH, SOUTH, EAST, WEST, or HD, D, CR, P, N, etc.

- May have other **fields**
- May have **methods**
- May use **constructors**
- Can be used as argument to **iterators**
 - use static `values()` method.