

# Java Tutoring Session 1

CS 5004

# Topics covered

## Programming fundamentals

- Variables (assignment, declaration) common variable types
- Conditionals, loops, arrays & arraylists
- Classes, methods and method signatures

## Object Oriented Programming basics

- Four key principles: Abstraction, encapsulation, polymorphism, and inheritance
- Generics
- Interfaces

## Git / GitHub / Command line

# Java Basics: Hello World

See code demo

# Object Oriented Programming Fundamentals

# Principles of object oriented programming

The four fundamental principles of OOP

1. Encapsulation
2. Abstraction
3. Inheritance
4. Polymorphism

The central component of object oriented programming is in the name - it's about objects

# What are objects and classes?

Classes are the fundamental building blocks in Java - they are like a blueprint for creating objects (instances of that class)

In Java, every source file can contain at most one public class (must match the name of the file)

Objects are entities that you can manipulate by calling one or more of its methods

- Let's look at the anatomy of an object
- Objects contain instance variables, constructors, and methods

# Instance Variables

Instance variables store data for an object (attributes, member variables)

Syntax:

## Syntax 3.1 Instance Variable Declaration

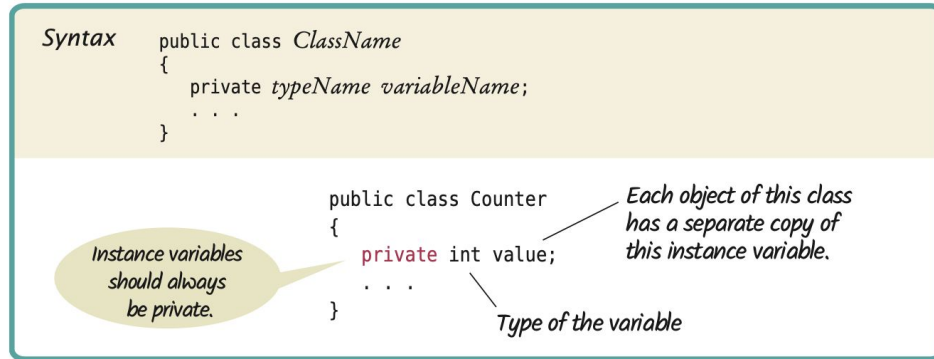


Image: Cay Horstmann

Access specifier, type of the instance variable, name of the instance variable

# Constructor

Very similar to a method but the name is always the same as the class and there is no return type (not even void!)

Its job is to initialize the instance variables of an object.

Can also overload constructors

## Syntax 2.3 Object Construction

Syntax `new ClassName(arguments)`

*The new expression yields an object.*

*Construction arguments*

`Rectangle box = new Rectangle(5, 10, 20, 30);`

*Usually, you save the constructed object in a variable.*

`System.out.println(new Rectangle());`

*You can also pass a constructed object to a method.*

*Supply the parentheses even when there are no arguments.*



# Methods

Accessor/Mutator methods (getter/setter)

Method signature is comprised of:

1. Scope (public vs private)
2. Static or non-static (invoked with or without an instance of a class)
3. Return type
4. Method name
5. Parameter list

# Person Class

Code demo

# Four Key Principles of OOP

# Encapsulation

Encapsulation is the process of hiding some of the implementation details and providing methods for accessing data

The idea that we are bundling together attributes and methods within a single unit

This allows for:

- Data hiding (internal state of an object is hidden from the outside world) we can only interact with the object through the well-defined methods
- Better code organization

# Abstraction

Process where you hide the implementation details from the user and only show the functionalities that are relevant to them

Separate between the interface and the implementation

Abstract methods in interfaces

# Inheritance

Inheritance is a mechanism wherein a new class is derived from an existing class. In Java, classes may inherit or acquire the properties and methods of other classes

Example: a bank account all has features but certain account types extended certain classes

# Polymorphism

The ability for the same entity (method or operator or object) to perform different operations in different scenarios, Java Generics

Example: overriding or overloading methods

# Command line fundamentals and Git

Mac	Windows
ls	dir
cd	cd
/	\
mdkir	mkdir