

Féidearthachtaí as Cuimse
Infinite Possibilities

Getting Started with OOP

W2-2025, Object-Oriented Programming, Week 2



Objectives

- What are the Fundamental concepts: classes, objects, attributes, methods.
- What do we mean by the Anatomy of a class: “constructors”, instance variables, methods.
- Putting theory into practice: Writing your first JAVA class and object.

Revision

- Basic syntax, see cheat sheet (C vs Java)
- Remember about the revision quiz.
- **Every week we are going to revise key concepts at the beginning of the lecture.**
- In the tutorials, I will:
 - Go through the **solution of the lab exercise**,
 - Revise important concepts from the lecture, and
 - Help you get comfortable with **Java code** and the core **Object-Oriented (OO) concepts**

Overview

What is a computer program?

Programming Language Evolution

- **Imperative programming**
 - Simple programs: **sequential lines of code**
- **Procedural programming**
 - Code is broken up into **functions** (sub-routines)
- **Object-Oriented programming**
 - Functions organized into **classes**
 - Classes represent objects in the real world

Key Concepts: Classes and Objects

- **Classes:** Blueprint or template for creating objects.
- **Objects:** Instances of classes with attributes and methods.
 - **Attributes:** Data or characteristics associated with an object.
 - **Methods:** Functions defined in classes to perform actions.

Object-oriented programming has four important pillars

ENCAPSULATION

ABSTRACTION

INHERITANCE

POLYMORPHISM



What is a class?

- A **class** is a blueprint from which individual objects are created (or, we can say a class is a data type of an object type). In Java, everything is related to classes and objects. Each class has its methods and attributes that can be accessed and manipulated through the objects.
- For example, if you want to create a class for *students*. In that case, "*Student*" will be a class, and student records (like *student1*, *student2*, etc) will be objects.
- We can also consider that class is a factory (user-defined blueprint) to produce objects.
 - (ref: https://www.tutorialspoint.com/java/java_object_classes.htm)

Objects and classes

Crucial to understand these

- A class is just the **template** In itself – (usually) doesn't do anything

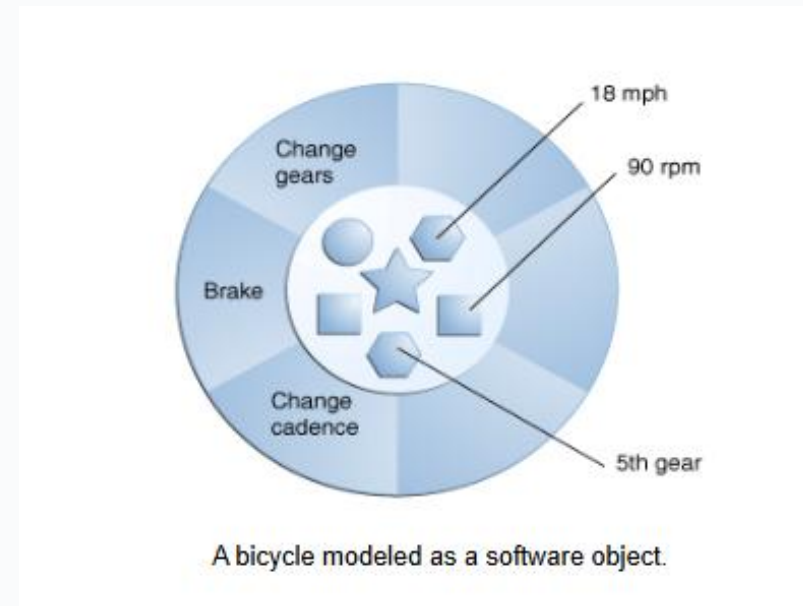
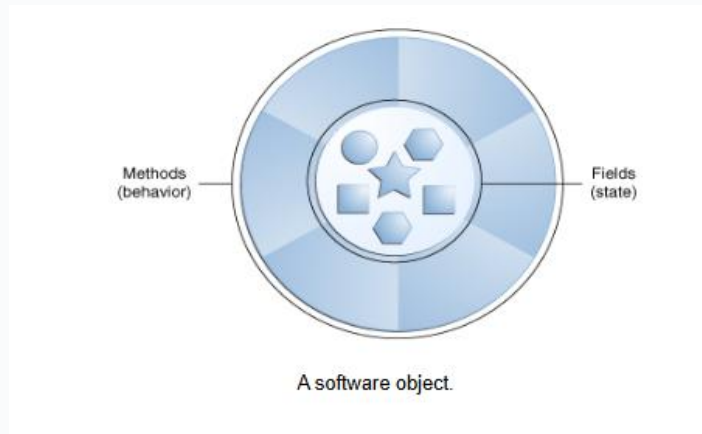
In the real world, you'll often find many individual objects all of the same kind. There may be thousands of other bicycles in existence, all of the same make and model. Each bicycle was built from the same set of blueprints and therefore contains the same components. In object-oriented terms, we say that your bicycle is an *instance* of the *class of objects* known as bicycles. A *class* is the blueprint from which individual objects are created

- To use, create “objects” from the class (called **instantiating** objects)
(what is the template for a book? (class) **BUT** if we give the book a value e.g. title it is an *object*)

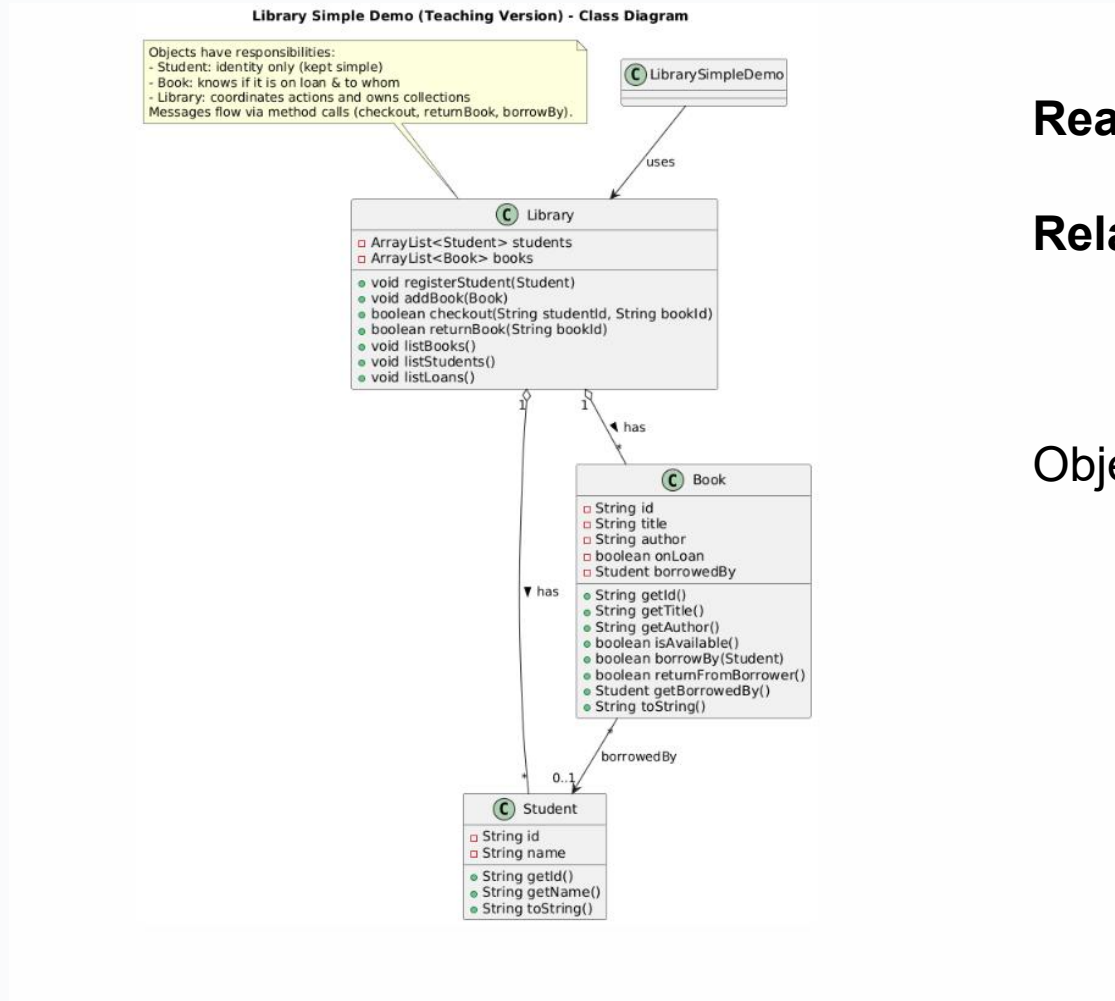
<https://docs.oracle.com/javase/tutorial/java/concepts/class.html>

What is an object ?

- Real-world objects share two characteristics: They all have *state* and *behavior*



Context : Class Demo



Real-world entities:

Student, Book, Library

Relationships:

Library has Students and Books.

Students borrow Books.

Objects interact to mirror the real-world library system.

Objects and Classes

- Use OOL to build a program – Create a model of some part of the world. – *simulating the real world in a computer*
- Java objects model objects from a problem domain. In the real world we interact with objects every day without thinking about how they work or why—we tell someone or something to do something and it does (mostly)
- This is going to seem strange – but when you code imagine the objects are alive and knows things --- don't worry I will explain what I mean

Put Simply

- In Object-Oriented Programming (OOP), an object is like a virtual thing in a computer program.
- Objects represent real-world entities or concepts.
- Each object has unique characteristics (attributes) and can perform certain actions (methods).
- Objects help organize code by modelling real-world relationships and interactions.

DEMO

```
1 package Friday;
2
3 public class Car {
4
5     // Attributes (fields)
6     private String model;           // declaration only
7     private int year = 2020;        // with initial value
8     public double price;            // accessible outside
9
10
11     // Method with no return
12     public void startEngine() {
13         System.out.println("Engine started.");
14     }
15
16     // Method with return value
17     public int getYear() {
18         return year; // returns attribute
19     }
20
21     // Method with parameters
22     public void setModel(String newModel) {
23         model = newModel;
24     }
25 }
```

10 min In-class Group Work Exercise



1. What **characteristics** have the following things?

2. What **behaviours** have the following **things**?

- Group 1: Car
- Group 2: Dog
- Group 3: Student
- Group 4: Book