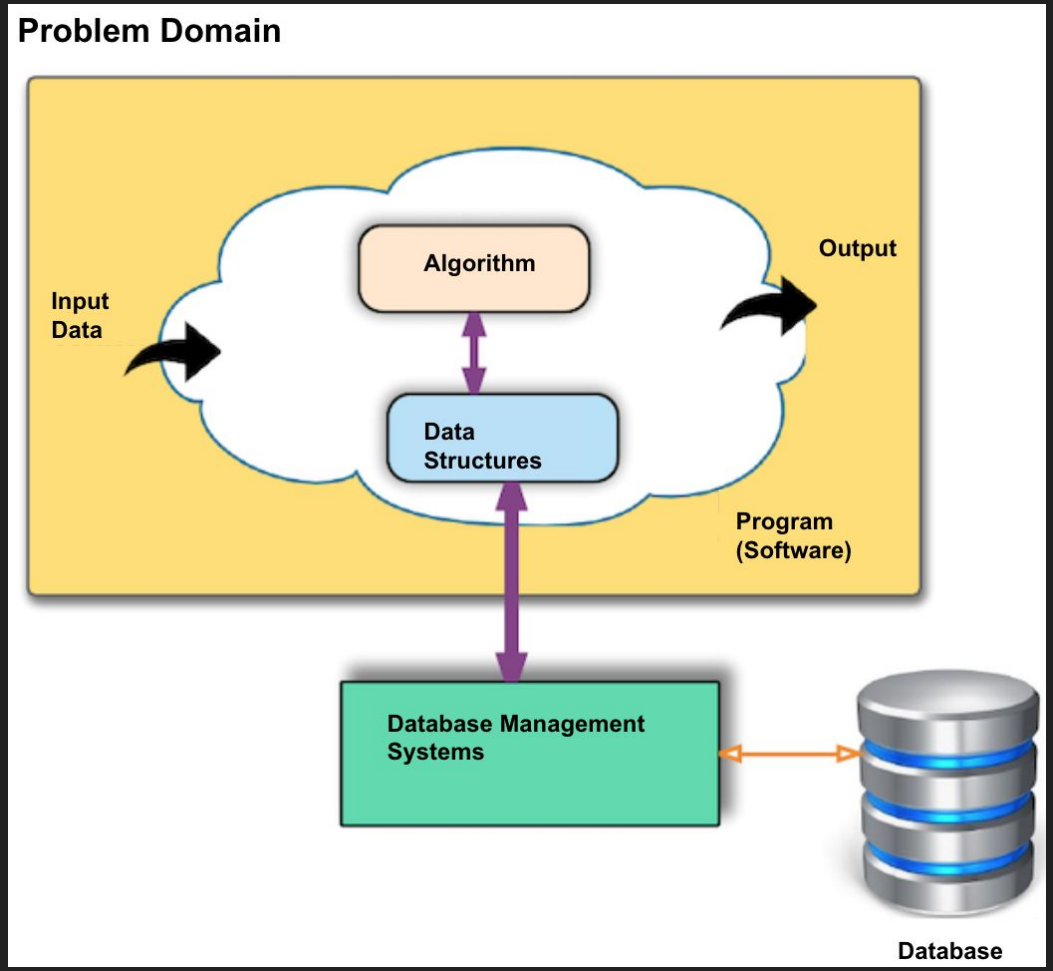
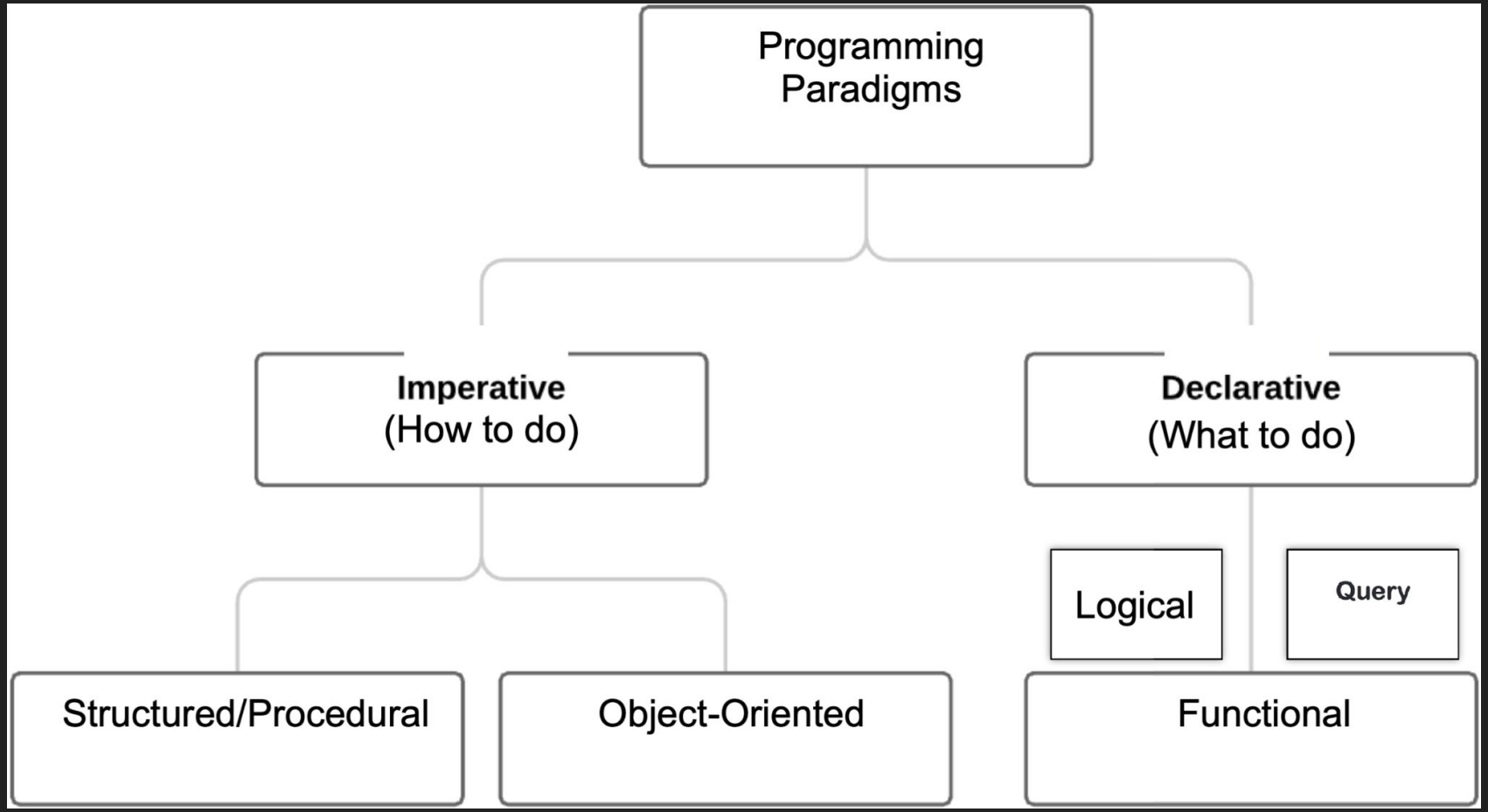
OOP

Introduction to Object-Oriented Programming

Programming: The Big Picture



Overview of Programming Paradigms

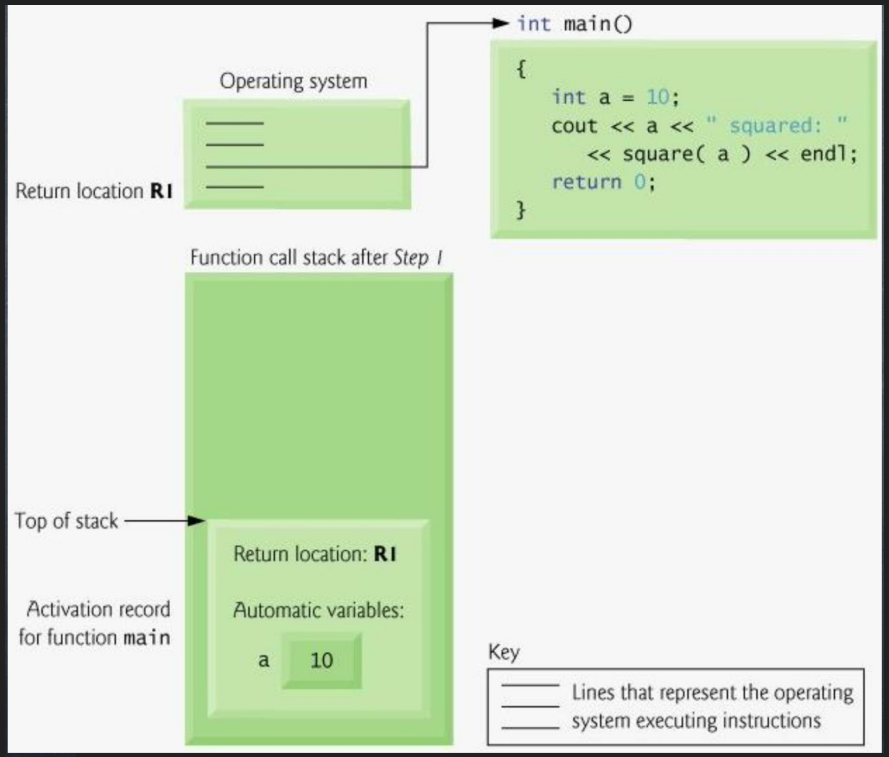


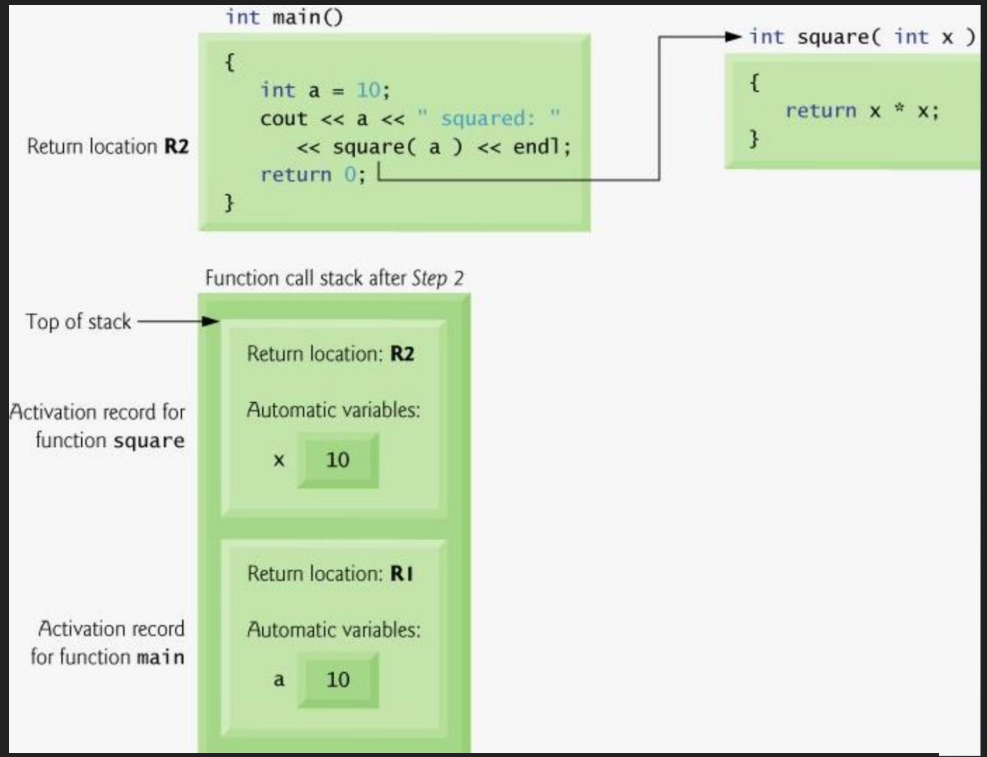
Structured Programming

Structured programming is a programming paradigm aimed at improving the clarity, quality, and development time of a computer program by making extensive use of subroutines, block structures and for and while loops in contrast to using simple tests and jumps such as the goto statement which could lead to "spaghetti code" which is difficult both to follow and to maintain.

https://en.wikipedia.org/wiki/Structured_programming

Structured Programming: Function Call





https://en.wikipedia.org/wiki/Structured_programming

Object-Oriented Programming

In structured programming, the focus is directly on the problem, and methods are developed to solve the problem (control-centric).

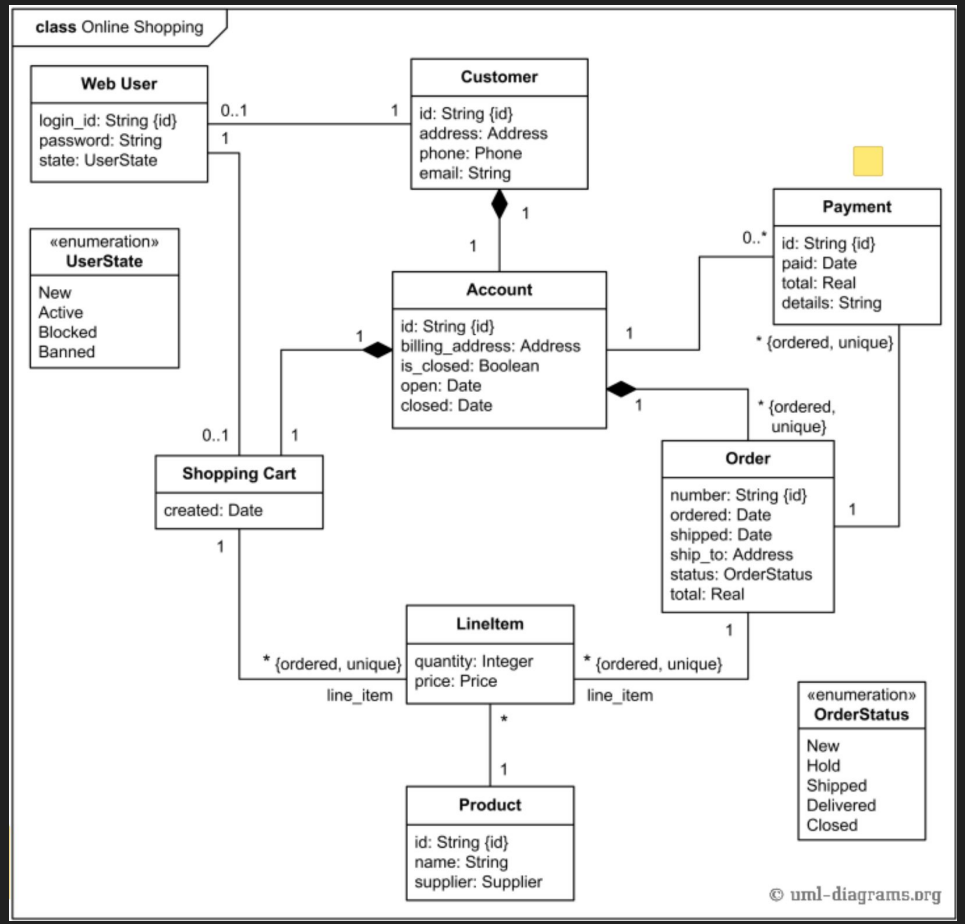
In object-oriented programming, however, the fundamental component is the object (data-centric), and programs are composed of interactions between objects. An object contains both data and the methods that process that data. Software developers concentrate on designing and developing the classes that construct these objects. This way, the program's logic is distributed across all modules in a data-centric manner.

Object-Oriented Programming

In structured programming, if a function requires data to perform a task, the necessary data is passed as parameters.

In object-oriented programming, however, the task to be performed is executed by the object, and functions can access data without the need to pass parameters.

Object-Oriented Programming



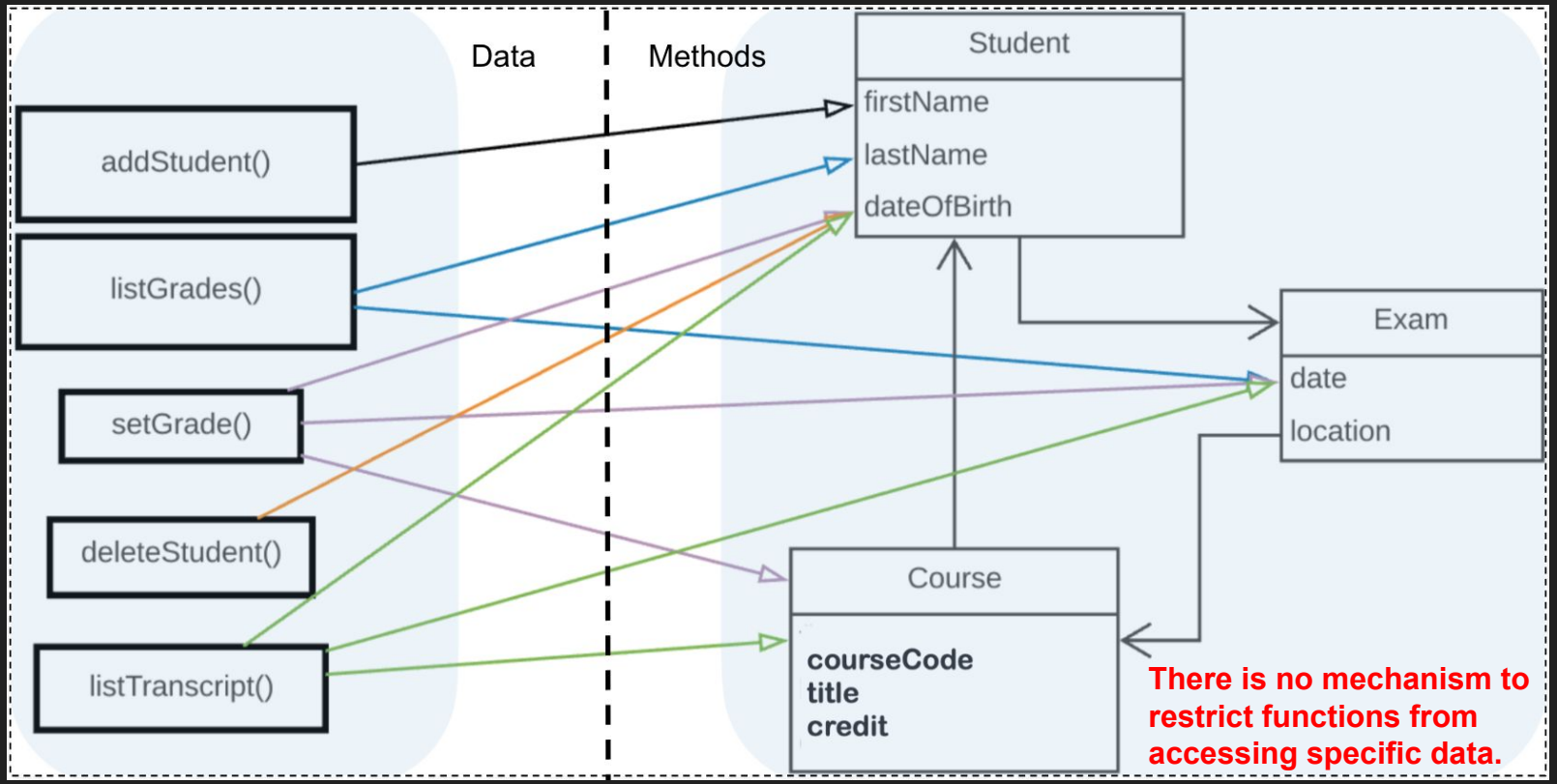
Object-Oriented Programming vs Structured Programming

As the system grows, the relationships become more complex.

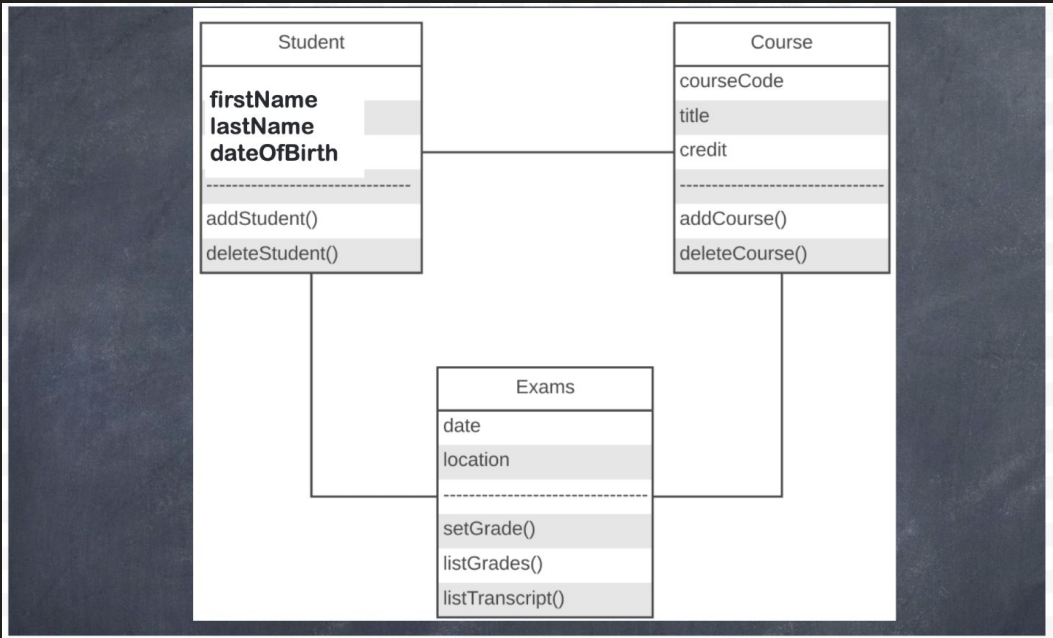
Making changes (adding, removing, or correcting) becomes increasingly difficult and can lead to unexpected side effects.

Object-Oriented Programming vs Structured Programming

We want to change the DOB field to 4 digits. The methods utilizing this data will also need to be updated. Which methods are accessing which data?



Object-Oriented Programming vs Structured Programming



High Coherence

high cohesion refers to how closely related and focused the responsibilities of a single class or module are. A class with high cohesion performs one well-defined task, making the code easier to understand, maintain, and modify.

Low Coupling

The connection between modules should be weak. This prevents changes from affecting related modules.

Key Properties of Object-Oriented Programming

- ❏ Encapsulation (Information Hiding)
- ❏ Inheritance
- ❏ Polymorphism
- ❏ Abstraction
- ❏ Modular Programming
- ❏ Code Reuse Maintenance
- ❏ Design Principles (SOLID)
- ❏ Design Patterns

https://github.com/celalceken/OOPCourseCodeSamples/tree/main/lecture1

Functional Programming (Lambda Statements)

- Programs are composed of the combination of pure functions.
  - The same output is produced for the same input (no side effects).
  - No shared/global variables exist.
  - The state of objects passed as parameters cannot be modified.
- No iterative structures (loops) are present.
- Since state changes are not allowed, parallel programs can be written (used in stream processing).

https://github.com/celalceken/OOPCourseCodeSamples/blob/main/lecture1/collections/Main.java