Lect 02
Today's Outline

- Programming Paradigms
- Problems with Procedural Programming
- Introduction to Object Oriented Programming
- Review

Programming Paradigms

A **programming paradigm** is a style, or "way," of **programming**.

Programming Paradigms

- Two broad groups
  - Traditional programming languages
    - Sequences of instructions
    - First, second and some third generation languages
  - Object-oriented languages
    - Objects are created rather than sequences of instructions
    - Some third generation, and fourth and fifth generation languages

PROGRAMMING PARADIGMS

- Structured Programming Paradigm
  - Procedural Programming
- Object Oriented Programming Paradigm

Structured Programming

- Structured programming is any programming when functionality is divided into units like for loop, while loop, if... then etc block structure.
- Using function
- Function & program is divided into modules
- Every module has its own data and function which can be called by other modules.
- Top down approach

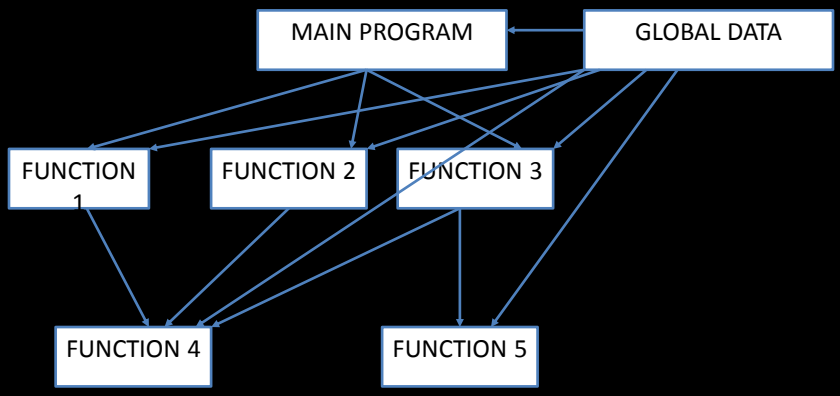
30

Procedural Programming Paradigm

- A procedural programming language consists of a set of procedure calls and a set of code for each procedure.

- In Procedure Oriented Programming, importance is not given to **data** but to functions as well as **sequence** of actions to be done.

Procedural Programming



32

Problems with Procedural Programming

- Unrestricted Access
- Difficulty in modification
- Impossible Code Reuse
- No real world modeling
- Difficult to create new data types reduces extensibility
- Importance is given to the operation on data rather than the data.

Introduction to Object Oriented Programming

- *In Object Oriented Programming, the program is based on real world things (objects) designed and built to model how these "real" things interact.*

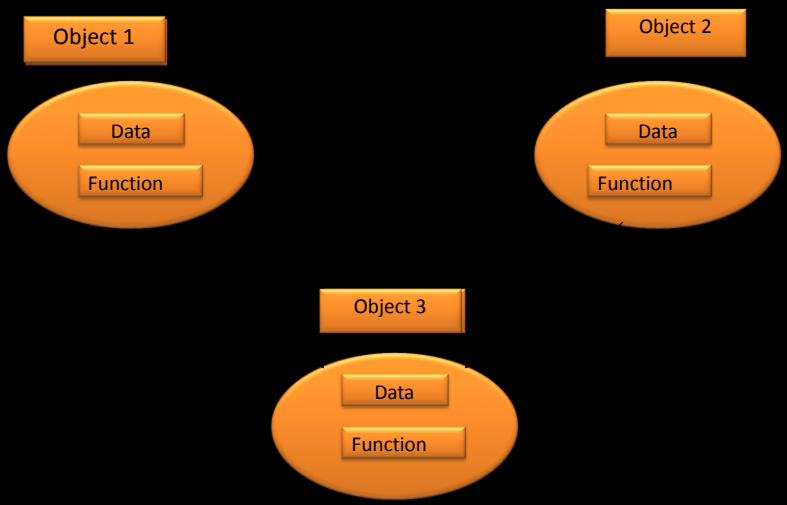
- Objects have both data and methods

- Key idea in object-oriented:

The real world can be accurately described as a collection of objects that interact.

35

OBJECT ORIENTED PROGRAMMING

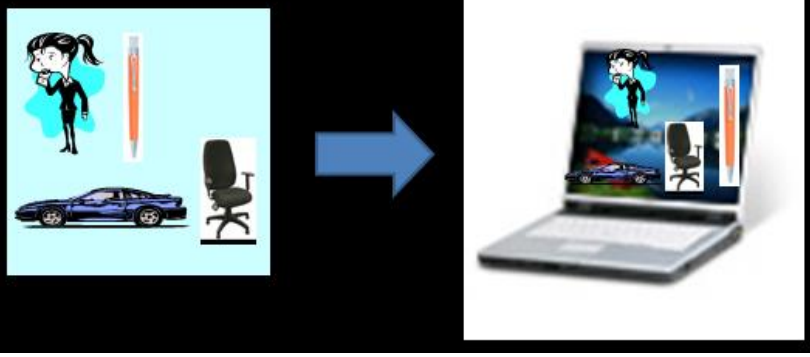


36

What is Object-Orientation?

- What is Object?

- An "object" is anything to which a concept applies, ***in our awareness***
- Things drawn from the problem domain to the solution space.
  - E.g., a living person in the problem domain, a software component in the solution space.



What is Object-Orientation?

- A technique for system modeling
- OO model consists of several interacting objects

What is a Model?

- A model is an abstraction of something
- Purpose is to understand the product before developing it

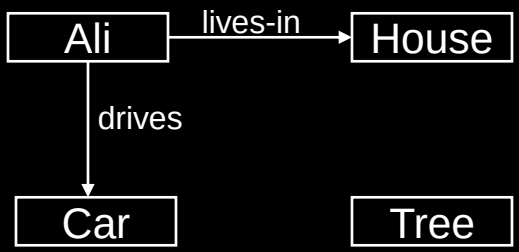
Examples – Model

- Highway maps
- Architectural models
- Mechanical models

Example – OO Model

…Example – OO Model

- Objects
  - Ali
  - House
  - Car
  - Tree
- Interactions
  - Ali lives in the house
  - Ali drives the car



Why OOP?

Benefits of using OOP Paradigm

- OO models map to reality
- Improved software-development productivity
- Improved software maintainability
- Faster development
- Lower cost of development
- Higher-quality software

Imperative Vs Declarative Programming Paradigms

Imperative Programming

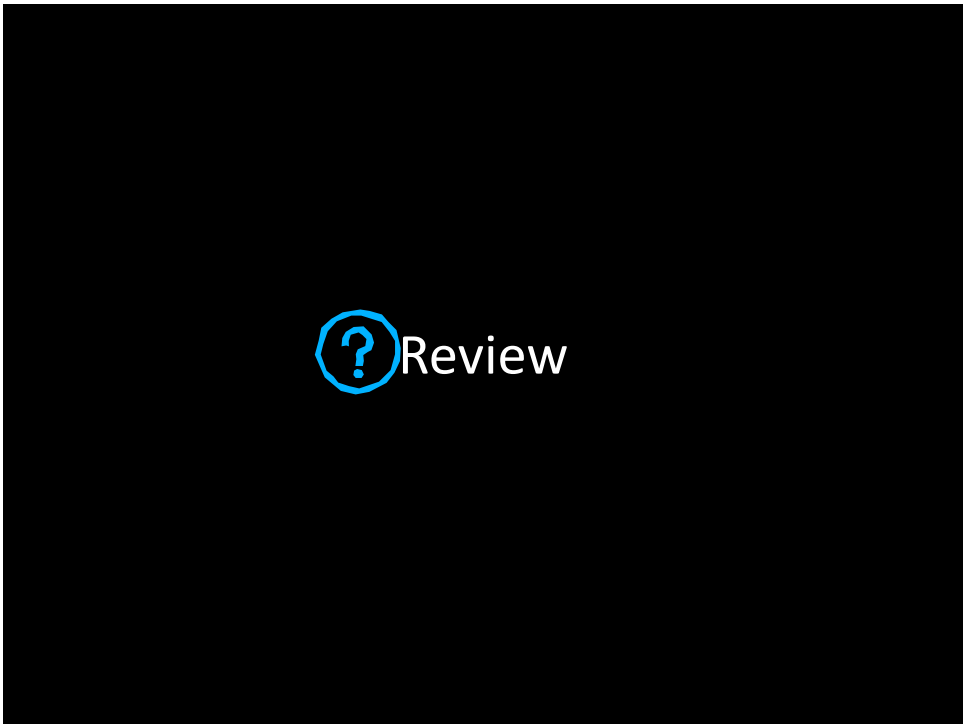
- With **imperative** programming, you tell the compiler what you want to happen, step by step.
- List<int> results = new List<int>(); foreach(var num in collection) { if (num % 2 != 0) results.Add(num); }

Declarative Programming

- With **declarative** programming, on the other hand, you write code that describes what you want, but not necessarily how to get it (declare your desired results, but not the step-by-step):
- var results = collection.Where(num => num % 2 != 0);

Imperative Vs Declarative Programming Paradigms

- Declarative programming is when you say *what* you want, and imperative language is when you say *how* to get what you want.


Review