COEN-244 Tutorial #2

OOP Overview

Object Oriented Programming is the process of defining your program/procedure in terms of objects :: objects give sensible functionality (complements) to the program

- Classes define the bases characteristics/behaviors of the objects
- 'Classes' in OOP refer to classification of a set of objects that have behave similarly and have homogeneous properties
- Advantages of OOP: Clearly Structured Code, Keeps code DRY (Don't Repeat Yourself):
 easier to Modify, Debug & Maintain
- OOP is heavily used in implementing DSA
- It also helps with writing less comments!

Things to know before going ahead

- Differences between classes and objects(instances)
- Attributes, Methods, Constructors, Destructors, and Encapsulation
- Encapsulation: Public, private, and protected
- Accessors (getters) and mutators (setters)
- Code organization separate .h and .cpp files
- Static variables how to give special IDs to objects
- Member initialization list
- C++ Structures (Structs)

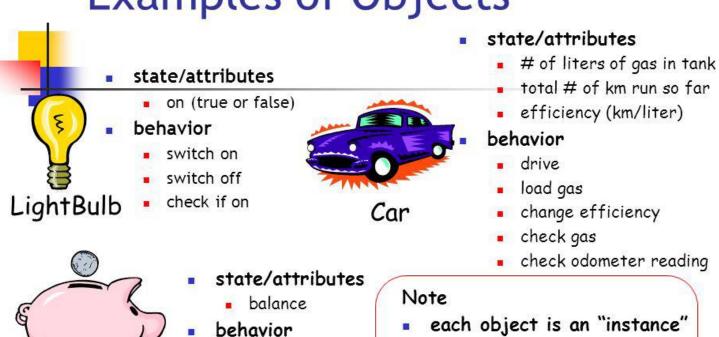
Some OOP Concepts

- Data Abstraction: Showing only what is necessary
- Encapsulation: Giving the right permissions
- Data members: Defining the objects
- Use of Pointers: Dealing with pointers in OOP
- Functions & Objects: Use-cases of objects in procedural funcs

Let's link all this together, shall we?!

Examples of OOP

Examples of Objects



BankAccount

- - deposit
 - withdraw
 - check balance
- of that "class" of object
- each instance has its own values for its attributes
 - e.g., different accounts can have different balances

THANK YOU