OOP: OBJECT ORIENTED PROGRAMMING

ENG. ISURU SHANAKA LAKMAL

Introduction

- Object Oriented programming (OOP) is a programming paradigm that relies on the **concept of classes and objects**.

- It is used to **structure a software program** into **simple, reusable pieces of code blueprints** (usually called classes), which are used to create individual instances of objects.

- There are many object-oriented programming languages including JavaScript, C++, C#, Java, and Python.

Introduction

Class

- A class is an **abstract blueprint** used to create more specific, concrete objects.
- Classes often represent broad categories, like Car or Dog that share attributes.
- These classes define **what attributes an instance of this type will have, like color**,
- But **not the value of those attributes** for a specific object.

Introduction

Class…cont.

- Classes **can also contain functions, called methods** available only to objects of that type.

- These **functions are defined within the class** and perform some action helpful to that specific type of object.

Principles of OOP

- Encapsulation
- Inheritance
- Polymorphism
- Abstraction

Principles of OOP

Encapsulation

- Encapsulation refers to the creation of **self-contained modules (classes)** that bind processing functions to its data members.
- The **data within each class is kept private**.
- **Each class defines rules** for what is publicly visible and what modifications are allowed.

Principles of OOP

Inheritance

- Classes may be created in hierarchies, and **inheritance lets the structure and methods in one class pass down the class hierarchy**.

- By inheriting code, complex behaviors emerge through the reuse of code in a parent class.

Principles of OOP

Inheritance …cont.

- If a step is added at the bottom of a hierarchy, **only the processing and data associated with that unique step must be added**.

- Everything else **above that step may be inherited**.

- **Reuse is considered a major advantage** of object orientation.

Principles of OOP

Polymorphism

- Object oriented programming **lets programmers create procedures for objects** whose exact type is not known until runtime.

- For example, a screen cursor may change its shape from an arrow to a line depending on the program mode.
- The routine to move the cursor on screen in response to mouse movement can be written for “cursor”,
- And polymorphism lets the right version for the given shape be called.

Principles of OOP

Abstraction

- An abstraction denotes the **essential characteristics of an object**,
- That **distinguish it from all other kinds of objects**
- And thus provide **crisply defined conceptual boundaries**, relative to the perspective of the viewer.

Principles of OOP

Abstraction …cont.

- Abstraction **denotes a model, a view, or some other focused representation for an actual item**.

- It’s the **development of a software object to represent an object** we can find in the real world.

- **Encapsulation hides the details of that implementation**.