Title	:	OOPS Realtime example
Author	:	Sibi Senthilkumar
Created On	:	19.11.2024
Last Modified Date	:	21.11.2024
Reviewed By	:	
Reviewed On	:	

OOPS – Object Oriented Programming Language

The concept of binding values and functions under a single user-defined datatype 'Class'

Class – It is a user–defined layout that describes what a specific kind of object look like. In other words, it is a blueprint of an object which consists of states and behaviors (i.e., data and functions).

Object – It is variable to access the member data and member functions. In other words, it is an instance of a class.

Need for OOPS: Object packs data and the operation ensures that only operations are publicly available and internal details of data structures are hidden.

Objects in Class use Heap Memory whereas Objects in Struct use Stack Memory. Objects that use Heap memory are **Reference Types**. This helps updating the details through reference which reduces memory footprints.

Attributes – Variables inside the class that stores data used in the application.

Method – Functions inside the class that are used to express the behavior of the object.

Constructor – It is used to initialize instance members of the class.

Attributes inside constructor are referred as Instance Attributes.

Attributes outside constructor are referred as Class Attributes.

Concepts in OOPS:

Encapsulation – It protects the class from accidental changes or deletion and promotes code reusability. It wraps up the data.

It is achieved using Access Modifiers(Public, Protected and Private).

<u>Example</u>: In game stats, cash, health points are only visible but not allowed to change by the users. This ensures that encapsulation has been used there.

Abstraction – It serves as a blueprint for other classes which can then define their own specific behaviors. An abstract class can only be inherited by sub-classes. It wraps up the code.

<u>Example</u>: During game development, an abstract class called 'Character' is created which includes all the basic characteristics of the character. Each type of character (e.g Male: Hawkeye, Ironman, Cap and Female: Scarlet, Gamora) can inherit 'Character' class and define their own specific behaviors and methods.

Polymorphism – Polymorphism allows different objects to respond to the same method or functions in different ways. Polymorphism works by using inheritance and abstraction.

It comprises of Runtime and Compile Time Polymorphism.

Runtime Polymorphism is implemented using method overriding(Multiple Methods with same name but different parameter and return type) and Compile Time Polymorphism is achieved using method overloading(same method name and parameters in different class).

Example: Attacking style differs from every character but the action still remains 'Attack' [method].

Inheritance – It allows one class to derive the properties or use the properties and methods from another class. The class that inherits is called the 'child' or 'subclass' and the class it inherits from is called 'parent' or 'superclass'.

Example: Every characters have ID, Name, Special Power, Attack Damage, Defense Damage etc., which can be inherited and modified according to the character.

Reference:

1. Data Camp
2. Linkedin
3. Enjoy Algorithm