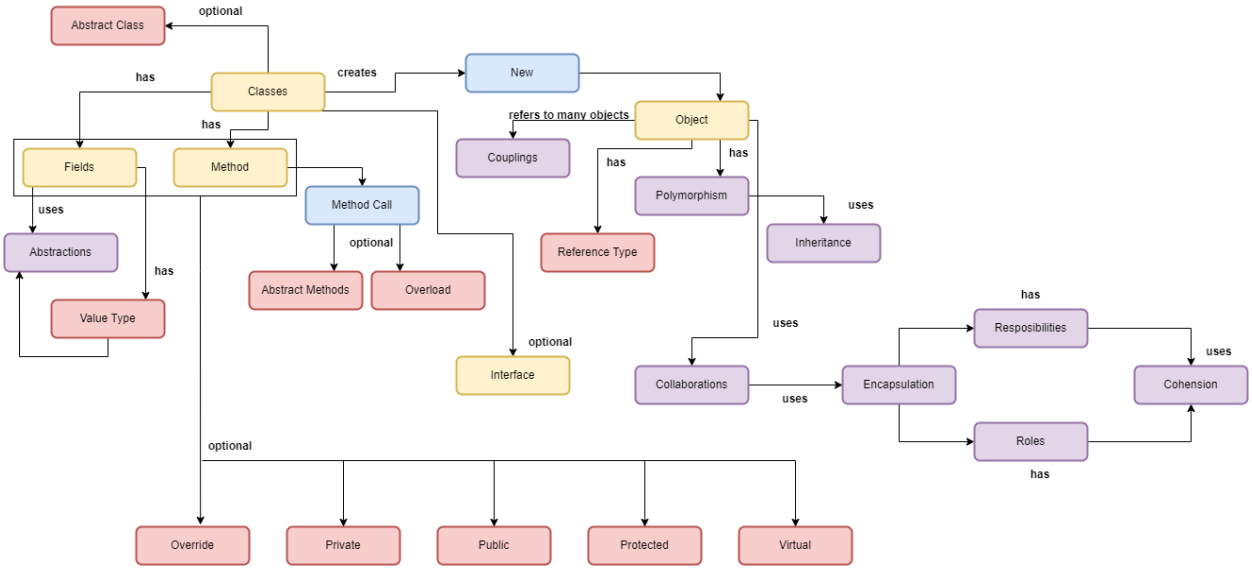
Task 7.1 – OOP Concepts



Concepts:

Abstractions –

Abstractions is an Object-Oriented Programming Concept that implements the use of needed attributes and privates unnecessarily data to interpret real-world concepts in different forms of representation. In addition, abstraction or the abstract method that are used on classes to create abstract classes that are intended to be used as a base class for other classes extend functionality.

Encapsulations –

Encapsulation is an Object-Oriented Programming Concept which is grouping data types and methods that functions that are used in classes which are encapsulated to keep the data within the scope.

Inheritance –

Inherits functions and methods from parent classes to create a new object in child classes. Moreover, Inheritance is used to model generalization and specialization in Object Oriented Code as it helps your code to be more flexible, extensible and adaptable. For Example, In the Shape Drawer task we used the shape class to act as a parent class to make more shapes such as a Rectangle and Circle class while implementing special features for each of those classes.

Polymorphism –

Polymorphism is an Object-Oriented Programming Concept where it is used to create multiple types of objects for different object classes using one class as the blueprint to implement these mechanisms and can also be redefined if needed such as overriding methods using features like override and virtual inside the base class.

Roles –

set of responsibilities that implemented into objects.

Responsibilities –

Responsibilities are conditions described to perform a task and compromises.

Collaborations –

Collaboration is the interaction of objects and roles to solve the issue of objects not performing responsibilities properly when not implemented properly. Therefore, it is very important to have proper collaboration and there are 3 types of collaborations such as Association, Aggregation and Dependence.

Coupling –

Coupling is the concept of having an Object-Oriented Design where classes and methods being connected closely and depended on each other.

Cohesion –

Cohesion is the design implementation of a class where high cohesion means they include limited number of unnecessary actions or tasks meaning these classes have a better Object-Oriented Design whereas if Low Cohesion means the class has not been designed well.

Artefacts:

Class –

Class in Object-Oriented Programming is a blueprint used to create objects that provide various attributes and implement behaviors such as methods and functions.

Object –

Object in Object-Oriented Programming is a data structure class that contains components such as methods and is used in other classes to execute or call the various define functions within the object.

Interface –

Interface is an Object-Oriented Programming Concept that uses a simple syntax to define functions that specify where these lists on functions can be implemented in classes. Moreover, Classes can inherit from one class but can implement many interfaces as it allows you to access features in a flexible way.

Method –

Method is a type of action in the Object-Oriented Programing framework which performs various types of operations or functions within a class. Furthermore, Methods can accepts manipulate by accepting parameters as arguments and provide an output when it is called by an object or class.

Fields –

Fields in Object-Oriented Programming are variables that are data encapsulated by other classes or objects that have different types of data fields that can be static, instance or both.

Action:

Method Call –

Method Call is when methods that have a defined behavior to output is being called from existing classes to perform a task.

New –

New in Object-Oriented Programming is a allows to create a new user-defined object type of an already existing object.

Terminology:

Value Type –

Value Type is memory pointer that is used in Object-Oriented Programming as a way of storing data on a stack to allow creation of flexible relationships between objects.

Reference Type –

Reference is another form of memory pointer in Object-Oriented Programming that refers to an instance in a class or an object.

Abstract Class –

Abstract Class are subclasses that contain abstracted methods that declared but have no implementation and depends on child classes.

Abstract Methods –

Abstract Methods are methods that is defined but has no implementation but can be redefined by child classes to implement tasks for these methods.

Private –

Private is where only classes that contain the fields have access to it.

Public –

Public is where any class can refer to the fields or methods.

Protected –

Protected is where only current or sub classes have access to these fields or methods.

Overload –

Overload is where it allows to declare same methods with same name but different conditions/functions.

Override –

Override is where it allows child class to declare different implementation for a function or methods from its parent class.

Virtual –

Virtual is a function that is implemented to be able to allow functions to be redefined and extended in other classes.