

assignment4

April 16, 2023

0.1 QUESTION 1

- 0.2 ANS - In object-oriented programming, a class is a blueprint or template for creating objects that define a set of properties and methods. An object, on the other hand, is an instance of a class. It is a runtime entity that represents a real-world entity, and has its own set of attributes and behaviors defined by the class it belongs to.
- 0.3 For example, let's consider a class called Car. The Car class may have attributes such as make, model, year, color, price, and methods such as start, stop, accelerate, and brake. These attributes and methods define the characteristics and behaviors of a car.

0.4 QUESTION 2

- 0.5 ANS- 1) Encapsulation: Encapsulation refers to the practice of hiding the internal details of an object and exposing only what is necessary. It involves bundling data and methods that operate on that data into a single unit, and restricting access to the data from outside the unit. Encapsulation helps to protect the integrity of an object's data, and promotes modularity and reusability.
- 0.7 2) Inheritance: Inheritance allows one class to inherit the properties and methods of another class. The class that is being inherited from is called the superclass, and the class that inherits from it is called the subclass. Inheritance promotes code reuse and allows for the creation of hierarchies of classes with shared characteristics and behaviors.
- 0.8 3) Polymorphism: Polymorphism means "many forms" and refers to the ability of objects to take on different forms or behave in different ways depending on the context in which they are used. In OOP, polymorphism is achieved through method overloading and method overriding.
- 0.9 4) Abstraction: Abstraction refers to the practice of representing complex real-world entities as simplified models in code. It involves identifying the most important attributes and behaviors of an entity and representing them in a way that is easy to understand and work with. Abstraction helps to reduce complexity and increase efficiency in software development.

0.10 QUESTION 3

- 0.11 init() function is a special method that is used to initialize an object of a class. It is called a constructor because it is automatically called when an object is created. The purpose of the init() function is to set the initial values of the attributes of the object.
- 0.12 For example, let's consider a class called Person. The Person class may have attributes such as name, age, and gender. To initialize these attributes, we can define the init() function like this:

```
[2]: class Person:
      def __init__(self, name, age, gender):
          self.name = name
          self.age = age
          self.gender = gender
```

0.13 QUESTION 4

0.14 self is a special keyword that is used as a reference to the object that is being created or manipulated. It is used to access the attributes and methods of the object within the class definition.

0.15 QUESTION 5

0.16 Inheritance is a fundamental concept in object-oriented programming (OOP) that allows one class to inherit properties and methods from another class. The class that is being inherited from is called the superclass, and the class that inherits from it is called the subclass.

0.17 Single inheritance: A class is derived from only one base class.

0.18 Multiple inheritance: A class is derived from more than one base class.

0.19 Multilevel inheritance: A class is derived from a base class, which in turn is derived from another base class.

0.20 Hierarchical Inheritance: Multiple classes are derived from a single base class.

0.21 Hybrid inheritance: It is a combination of more than one type of inheritance, such as combining multiple inheritance and multilevel inheritance.

[]:

[]:

[]:

[]:

[]:

[]:

[]:

[]:

[]: