Assignment 09: 08 Feb 2023

Q1. What is Abstraction in OOps? Explain with an example.
Q2. Differentiate between Abstraction and Encapsulation. Explain with an example.
Q3. What is the abc module in python? Why is it used?
Q4. How can we achieve data abstraction?
Q5. Can we create an instance of an abstract class? Explain your answer

Q1. What is Abstraction in OOps? Explain with an example.

Ans: Abstraction in python is defined as **a process of handling complexity by hiding unnecessary information from the user**.
Eg-
Consider a **real-life example of a man driving a car**. The man only knows that pressing the accelerator will increase the speed of the car or applying brakes will stop the car but he does not know how on pressing the accelerator the speed is actually increasing, he does not know about the inner mechanism of the car or the implementation of the accelerator, brakes, etc in the car. This is what abstraction is.

Q2. Differentiate between Abstraction and Encapsulation. Explain with an example.

Ans: Just in one line you say, Abstraction is hiding the details and implementation of the code. Encapsulation is hiding the data and controlling the visibility of the code.

S.No.	Abstraction	Encapsulation
1	It solves the problem at the design level.	It solves the problem at the implementation level.
2	It is used for hiding the unwanted data & giving relevant data.	It means hiding the code & data into a single unit to protect them from the outside world.
3	It lets you focus on what the object does instead of how it does it.	It means hiding the internal details or mechanism of how an object does something.
4	Outer layout, used in terms of design. Eg- outer look of a mobile phone like it has a display screen & keypad buttons to dial a number.	Inner layout, used in terms of implementation. Eg- inner implementation details of a mobile phone, how keypad button & display. The Screens are connected with each other using circuits.

Q3. What is the abc module in python? Why is it used?

Ans: The 'abc' (abstract base class) module in Python library provides the infrastructure for defining custom abstract base classes.
It is used because of following reason :-

- It provides a way to define abstract classes in python, which are classes that cannot be instantiated on their own, but only serve as a base for other classes.
- to define abstract classes in python

- Python does not have native support for abstract classes.

Q4. How can we achieve data abstraction?

Ans: In Python, abstraction can be achieved by **using abstract classes and methods in our programs**.

Q5. Can we create an instance of an abstract class? Explain your answer

Ans: Abstract classes are not complete, as they may have some methods that are not defined. So **we cannot create an instance (or object) of an abstract class in Python.**