**ASSIGNMENT 06**

QUESTION 01:

OBJECT ORIENTED PROGRAMMING:

The concept of OOP in Python focuses on creating reusable code. This concept is also known as DRY (Don't Repeat Yourself). A process of using details from a new class without modifying existing class. Hiding the private details of a class from other objects.

QUESTION 02:

**Advantages of OOP:**

* It provides a clear modular structure for programs which makes it good for defining abstract datatypes in which implementation details are hidden.
* Objects can also be reused within an across applications. ...
* It makes software easier to maintain. ...
* Reuse also enables faster development.

QUESTION 03:

DF BW FUNCTION AND METHOD:

Functions are called independently.functions are defined in structural language and methods are defined in object oriented languages. The difference between both is given below .Function are self contained program they have return some values and the functions are defined in structured languages like Pascal,C.

QUESTION 04:

CLASS:

In object-oriented programming, a class is a blueprint for creating objects (a particular data structure), providing initial values for state (member variables or attributes), and implementations of behavior (member functions or methods). The user-defined objects are created using the class keyword.

OBJECT:

In object-oriented programming (OOP), objects are the things you think about first in designing a program and they are also the units of code that are eventually derived from the process. ... Each object is an instance of a particular class or subclass with the class's own methods or procedures and data variables.

ATTRIBUTE:

In Object-oriented programming(OOP), classes and objects have attributes. Attributes are data stored inside a class or instance and represent the state or quality of the class or instance. ... One can think of attributes as noun or adjective, while methods are the verb of the class.

BEHAVIOUR:

Behavior. A class's behavior determines how an instance of that class operates; for example, how it will "react" if asked to do something by another class or object or if its internal state changes. Behavior is the only way objects can do anything to themselves or have anything done to them.