Assignment # 6   
Question 1:

Object oriented programming is a programming language model that organizes software design around data or objects, rather than functions and logic. An object can be defined as a data field that has unique attributes and behavior.    
  
   
Question 2:

The advantages of OOP are mentioned below:

* OOP provides a clear modular structure for programs.
* It is good for defining abstract data types.
* Implementation details are hidden from other modules and other modules has a clearly defined interface.
* It is easy to maintain and modify existing code as new objects can be created with small differences to existing ones.
* objects, methods, instance, message passing, inheritance are some important properties provided by these particular languages
* It implements real life scenario.
* In OOP, programmer not only defines data types but also deals with operations applied for data structures.

Question no 3:

Method is from the object oriented paradigm and refers to a function that is part of a class, so if the function is part of a class it can be referred to either as a method or a function. If the function is not a part of a class it should not be referred to as a method.

Question no 4:  
Class:  
A class is a blueprint for how an object is built, as well as being a sort of “parent category” for objects. Using the previous example, a class dictates the concept of a car—four wheels, an engine, a body, brakes, etc. It allows certain set criteria to be passed down to all objects in the class. All varieties of cars behave relatively the same on a basic level, but its their attributes and methods that make them unique.

Object :  
An object is the core unit of OOP. Objects are uniquely named and represent an instance of a class. Each object houses different states (attributes), and shared behaviors, called methods. For example, a Prius is an object in the class of “cars,” in a subclass of “hybrid cars.” Its attributes include anything from the number of doors it has to how its electric component is charged. It’s similar to other cars by its behavior—it drives—but its attributes are what set it apart.

Attribute:

Everything in Python is an object, and almost everything has attributes and methods. In python, functions too are objects. So they have attributes like other objects. All functions have a built-in attribute \_\_doc\_\_, which returns the doc string defined in the function source code. We can also assign new attributes to them, as well as retrieve the values of those attributes.

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. For example, to go back to the theoretical car class, here are some behaviors that the car class might have:

* Start the engine
* Stop the engine
* Speed up
* Change gear
* Stall