QUESTION # 1

**OBJECT ORIENTED PROGRAMMING**

OOP is a paradigm in which real world objects are each viewed as separate entities having their own state, which is modified only by built in procedures, called methods.

QUESTION # 2

**BENIFITS OF OOP**

1. Re-usability: Code can be re-used which is also known as DRY (Don’t Repeat Yourself)
2. Simplicity: software objects model real world objects, so the complexity is reduced, and the program structure is very clear
3. Modularity for easier trouble shooting: each object forms a separate entity whose internal workings are decoupled from other parts of the system;
4. Extensibility: adding new features or responding to changing operating environments can be solved by introducing a few new objects and modifying some existing ones;
5. Maintainability: objects can be maintained separately, making locating and fixing problems easier;

QUESTION # 3

**DIFFERENTIATE BETWEEN FUNCTION AND METHOD**

Method is called by its name, but it is associated to an object (dependent). A method is implicitly passed the object on which it is invoked. It may or may not return any data. A method can operate on the data (instance variables) that is contained by the corresponding class

Simply, function and method both look similar as they perform in almost similar way, but the key difference is the concept of ‘Class and its Object‘. Functions can be called only by its name, as it is defined independently. But methods can’t be called by its name only, we need to invoke the class by a reference of that class in which it is defined, i.e. method is defined within a class and hence they are dependent on that class

QUESTION # 4

**class**  
It is a template or blueprint about the capability of what an object can do. We can think of class as a sketch of a parrot with labels. It contains all the details about the name, colors, size etc. Based on these descriptions, we can study about the parrot. Here, parrot is an object. The example for class of parrot can be:

**Object**  
An object (Instance) is an instantiation of a class. When class is defined, only the description for the object is defined. Therefore, no memory or storage is located

**Attribute**  
In Object-oriented programming (OOP), attributes are the properties of the object. Attributes are data stored inside a class or instance and represent the state or quality of the class or instance. In short, attributes store information about the instance.

**Behavior**  
The behavior of an object is defined by its methods, which are the functions and subroutines defined within the object class. Methods determine what type of functionality a class has, how it modifies its data, and its overall behavior

QUESTION # 5