QUESTION NO1  
**OBJECT ORIENTED PROGRAMMING**

This Object-oriented programming (OOP) refers to a type of computer programming (software design) in which programmers define the data type of a data structure, and also the types of operations (functions) that can be applied to the data structure.In this way, the data structure becomes an object that includes both data and functions. In addition, programmers can create relationships between one object and another. For example, objects can inherit characteristics from other objects. For instance, an object could represent a person with a name property, age, address, etc., with behaviors like walking, talking, breathing, and running. Or an email with properties like recipient list, subject, body, etc., and behaviors like adding attachments and sending.

Put another way, object-oriented programming is an approach for modeling concrete, real-world things like cars as well as relations between things like companies and employees, students and teachers, etc. OOP models real-world entities as software objects, which have some data associated with them and can perform certain functions.

</blockquote>

QUESTION no 2

**BENIFITS OF OOP**

1. Simplicity: software objects model real world objects, so the complexity is reduced and the program structure is very clear
2. Modularity: each object forms a separate entity whose internal workings are decoupled from other parts of the system;
3. Modifiability: it is easy to make minor changes in the data representation or the procedures in an OO program. Changes inside a class do not affect any other part of a program, since the only public interface that the external world has to a class is through the use of methods;
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;
6. Re-usability: objects can be reused in different programs.

QUESTION3  
**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**  
A class is a blueprint for the object.We can think of class as an 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 :

**class Parrot:**

**pass**

**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 allocated.The example for object of parrot class can be:  
obj = Parrot()  
**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. In short, attributes store information about the instance.  
**Behaviour**  
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 NO 5

