

Q1: Define Object Oriented Programming Language?

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.

An **object-oriented programming language (OOPL)** is a high-level programming language based on the object-oriented model. To perform object-oriented programming, one needs an object-oriented programming language. Many modern programming languages are object-oriented, however some older programming languages, such as Pascal, do offer object-oriented versions. Examples of object-oriented programming languages include Java, C++ and Smalltalk.

Q2: List down the Benefits of OOP?

BENEFITS 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. The reuse of software also lowers the cost of development. More effort is put into the object-oriented analysis and design, which lowers the overall cost of development.
- It makes software **easier to maintain**. Since the design is modular, part of the system can be updated in case of issues without a need to make large-scale changes
- Reuse also enables **faster development**. Object-oriented programming languages come with rich libraries of objects, and code developed during projects is also reusable in future projects.
- It provides a good framework for code libraries where the supplied software components can be **easily adapted and modified by the programmer**. This is particularly useful for developing graphical user interfaces.
- **Better Productivity as** OOP techniques enforce rules on a programmer that, in the long run, help her get more work done; finished programs work better, have more features and are easier to read and maintain. OOP programmers take new and existing software objects and "stitch" them together to make new programs. Because object libraries contain many useful functions, software developers don't have to reinvent the wheel as often; more of their time goes into making the new program

Q3: Differentiate between function and method?

Functions:

1. Functions have independent existence means they can be defined outside of the class. Ex:- main() function in C, C++ Language
2. Functions are defined in structured languages like Pascal, C and object based language like JavaScript
3. Functions do not have any reference variable.
4. Functions are self describing unit of code.
5. All data that are passed to a function are explicitly passed.

Method:

1. Methods do not have independent existence they are always defined with in class. Ex:- main() method in C# Language that is defined with in a class
2. Methods are defined in object oriented languages like C#, Java
3. Methods are called using reference variable.
4. Methods are used to manipulate instance variable of a class.

Q4: Define the following terms:

1. Class

2. Object

3. Attribute

4. Behavior

CLASS: A class is an entity that determines how an object will behave and what the object will contain. In other words, it is a blueprint or a set of instruction to build a specific type of object.

OBJECT: An object is nothing but a self-contained component which consists of methods and properties to make a particular type of data useful. Object determines the behavior of the class. When you send a message to an object, you are asking the object to invoke or execute one of its method.

ATTRIBUTE:

Attributes are data stored inside a class or instance and represent the state or quality of the class or instance.

BEHAVIOR:

The behavior of an object is defined by its *methods*, which are the functions and subroutines defined within the object class. Without class methods, a class would simply be a structure.

Q5: Write a code in python in which create a class named it Car which have 5 attributes such like (model, color and name etc.) and 3 methods. And create 5 object instance from that class.

```
class Car():  
    def __init__(self,model,color,name,brand,capacity):  
        self.model=model  
        self.color=color  
        self.name=name  
        self.brand=brand  
        self.capacity=capacity  
    def speed(self):  
        print("speed is fast")  
    def start(self):  
        print ("Engine started")  
    def stop(self):  
        print ("Engine switched off")  
car_1=Car(108,"green","mehran","suzuki","6")  
car_2=Car(107,"black","mera","dihatsu","7")  
car_3=Car(109,"silver","margalla","suzuki","5")
```

```
car_4=Car(801,"silver","cultus","suzuki","5")
```

```
car_5=Car(200,"blue","swift","suzuki","5")
```