OOPs

Basic concepts of oops :

Object-oriented programming was invented to overcome the drawbacks of the pop. It employs the bottom-up programming approach. It treats data as a critical element in the program development and does not allow it to flow freely around the system. It ties data more closely to the functions that operate on it, and protects it from accidental modification from outside functions. Object-oriented programming as an approach that provides a way of modularizing programs by creating partitioned memory area for both data and functions that can be used as templates for creating copies of such modules on demand. Thus, an object is considered to be a partitioned area of computer memory that stores data and set of operations that can access the data.

Comparison of procedural programming and oop

POP	OOP
Employs top-down approach in program design.	Employs bottom-up approach in program design.
Data move openly around the system from function to function.	It treats data as a critical element in the program development and does not allow it to flow freely around the system.
Large programs are divided into smaller programs known as functions.	Programs are divided into what are known as objects.

Advantages of oop

1. The principle of data hiding helps the programmer to build secure programs that cannot be invaded by code in other parts of the program.
2. It is possible to have multiple instances of an object to co-exist without any interference.
3. It is easy to partition the work in a project based on objects.

OOP Languages

The languages should support several of the oop concepts to claim that they are object-oriented. Depending upon the features they support, they can be classified into the following two categories:

1. **Object-based programming languages,**
2. **Object-oriented programming languages**

Object-based programming is the style of programming that primarily supports encapsulation and object identity. Major features that are required for object based programming is:

1. **Data encapsulation**
2. **Data hiding and access mechanisms**
3. **Automatic initializations and clear-up objects**
4. **Operator Overloading**

Languages that support programming with objects are said to be object-based programming languages. They do not support inheritance and dynamic binding. Ada is a typical object-based programming language.

Object-oriented programming incorporates all of the object-based programming features, along with two additional features, namely, inheritance and dynamic binding. Object-oriented programming can therefore be characterized by the following statement:

Object-based features + inheritance + dynamic binding.

Languages that support these features include c++, small talk, object Pascal and Java.

Class

A class is a unit in which combines both data and the functions that operates on the data. The internal data of the class is called as member data and the functions are called member functions. The member functions mostly manipulate the internal data of a class. The variables of class are called objects or instances of a class.

Objects

Objects are the basic run-time entities in an object-oriented system. Programming problem is analyzed in terms of objects and the nature of communication between them. When a program is executed, the objects interact by sending message to one another. Each objects contains data, and code to manipulate the data.

Inheritance

Inheritance is the process by which objects of one class acquire the properties of objects of another class. In OOP, the concept of inheritance provides the idea of reusability. This means that we can add additional features to an existing class without modifying it. This is possible by deriving a new class from the existing one. The new class will have the combined features of both the classes.

Encapsulation

The wrapping up of data and functions into a single unit (called class) is known as encapsulation. Data encapsulation is the most striking features of a class. The data is not accessible to the outside world, and only those functions, which are wrapped in the class, can access it.

Operator Overloading

Operator overloading provides a flexible option for the creation of new definitions for most operators. We can almost create a new language of our own by the creative use of the function and operator overloading techniques. We can overload (give additional meaning to) all the operators except the following.

1. Class member access operators(.)
2. Scope resolution operator(::)
3. sizeof operator (sizeof)
4. Conditional operator(?:)

Dynamic binding

Binding refers to the linking of a procedure call to the code to be executed in response to the call. Dynamic binding (also known as late binding) means that the code associated with a given procedure call is not known until the time of the call at run-time.