P1

1.What is OOP

OOP (an acronym for -Object Oriented Programming) Object Oriented Programming is a programming method based on the concepts of classes and objects. OOP often focuses on manipulating objects rather than the logic to be able to manipulate them. OOP is a familiar foundation of today's design patterns. OOP aims to manage source code to increase reusability and, most importantly, to be able to encapsulate procedures with known properties through the use of objects.

2. OOP characteristics

2.1 Object-Oriented Development (OOD)

The focus on objects in object-oriented programming is its primary characteristic. When an object is defined by a developer, OOP is at its most fundamental. These objects can be used to create and communicate data, define business rules, and even manage the behavior of the programming environment. These objects are made by programmers using the OOD method. They develop a blueprint that specifies how these things will act in relation to one another. The interface is the name of this design. The methods that an object will be able to use in its interactions with other objects are specified in the blueprint. These techniques are frequently known as functions.

2.2 Encapsulation

OOP stresses the packaging of data and function inside objects as another aspect. Each object has a unique function that controls how other things interact with it. Data packing is a crucial component of OOP. This is so that hackers cannot alter the data that is kept in the database. A hacker's ability to change the data in a database could have catastrophic results for the company. The method through which the programmer separates data inside the object is called packaging. It stops hackers from examining database structure and data specifics. Abstraction and data packaging are the two key elements of the OOP model that increase security. They are necessary for storing sensitive data in any programming system.

2.3 abstraction

Reusable objects are being created using this technique in order to standardize traditional business logic. The application of objects is a crucial aspect of abstraction. They ought to be applied consistently in various applications. This facilitates logic standardization and makes it simpler to maintain across several programs. Abstraction makes it simpler for others to utilize OOP in their programs and simplifies the logic of standards. It promotes teamwork among several parties, making it simpler to develop applications that incorporate various datasets and services. Programs that are far more scalable and easier to maintain than conventional non-OOP applications can be made. But for this to happen, abstraction is required.

2.4 Classes and Objects

The next feature of Object-Oriented Programming is the definition of classes. A class is where the design for an object is determined. The blueprint defines methods, attributes, and other aspects of the object. Objects are created from layers. A class is also known as a blueprint for an object. It defines properties, methods, and other aspects related to the object. Objects are created from layers. A class is basically a blueprint for an object. It defines properties, methods, and other aspects related to the object.

2.5 Inheritance and Composition

The act of permitting one object to produce another is known as inheritance. The attributes, methods, and other features of another object can essentially be "inherited" by another object. The act of combining different objects to produce a brand-new, original object is known as composition. Programmers can essentially reuse the elements that are specific to other objects to construct new, original objects. One of the most significant characteristics of OOP is the combination of objects and the inheritance of their properties. It is the method used to generate and use objects in all computer languages. It serves as OOP's cornerstone.

2.6 Binding

In your application, binding is the process of connecting some characteristics of one object to another. For instance, you could link a text box to the button's text property so that if the button is chosen, the text box's value will be set to whatever is typed into it. Linking a form to a button will make it much simpler to process and debug your code when the code is executed on the server if you're constructing a web application and you want a form to submit data to your server. Dynamic binding and static binding are the two different types of binding. When the value of one object changes and as a result, the value of another object's property also changes, this is known as a dynamic constraint. When a property of an object is placed directly without modifying another property, this is known as static linking. Static links allow developers to accurately control what values are entered into each property, making it very simple for them to identify where their code is running in the app.

2.7 Message Passing

An essential component of object-oriented programming is the ability to transfer data inside a program. The fundamental tenet of data transmission is that each object keeps track of its own state and is capable of sharing it with other things. A simple way for an object to communicate with another object is to call the relevant method on the receiving object. This enables items to speak with one another and connect in intricate chains. Sending messages to other things is one way for objects to communicate. Sending a message from one item to another is fundamentally what it means to transmit a message. Each object has a unique "address" where other objects can communicate with it. When an object receives a message, it simply reads it and does the appropriate action based on it. This enables the construction of the full program from a collection of interconnected items. Programs become more complicated as a result of message transmission, and it will be much simpler for things to interact with one another in engaging ways. Even though messaging is one of the most fundamental kinds of communication in OOPS, it's a crucial idea to comprehend if you wish to create more intricate programs.

3. Types Of Relationships In Object Oriented Programming (OOP)