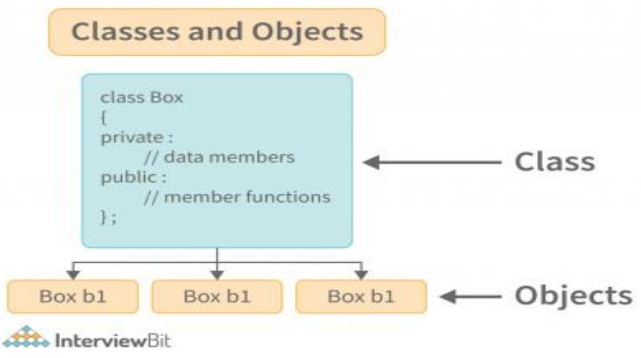
Characteristics Object oriented programming

- Class
- Objects
- Encapsulation
- Abstraction
- Polymorphism
- Inheritance
- Dynamic Binding
- Message Passing

Classes and Objects:

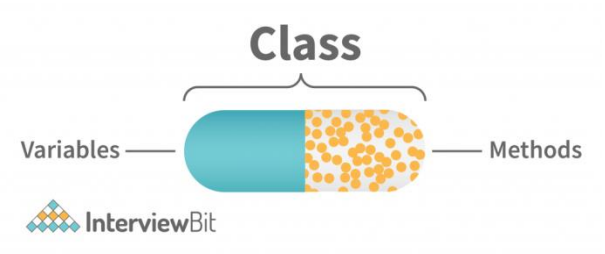
- A class is where the blueprint for an object is defined.
- The blueprint defines the methods, attributes, and other aspects of the object.
- Objects are created from classes.
- A class is also known as a blueprint for an object.
- It defines properties, methods, and other aspects associated with the object.



Encapsulation:

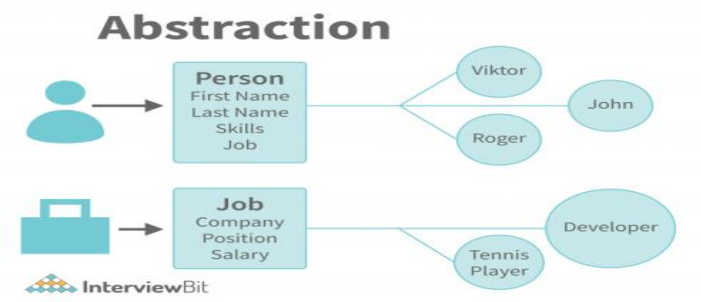
- functionality within objects. Objects are defined with specific functionality that governs the interactions between the objects

- This is because it prevents hackers from modifying data that is stored in the database
- Encapsulation is the process by which programmers isolate data within the object.



Abstraction:

- This is the process of creating reusable objects that are designed to standardize common business logic.
- It encourages collaboration between different parties, making it easier to create programs that integrate disparate data sets and functionality.
- It is possible to create applications that are much more scalable and maintainable than traditional non-OOP applications.



Inheritance and Composition:

- Inheritance is the process of allowing one object to create another.
- It essentially allows one object to "inherit" the properties, methods, and other aspects of another object.
- Composition is the process of combining multiple objects to create a new and unique object.
- It essentially allows programmers to reuse the elements that are specific to other objects to create a new and unique object.

Binding:

- Binding is the process of linking some property of an object to another object in your application.
- For example, you can bind a text box to the text property of a button so that when the button is selected, the text box's value gets set to whatever was entered into the text box.
- TEXT button will make it much easier to handle and debug your code as it's executed on the server.
- There are two types of binding: dynamic binding and static binding. Dynamic binding occurs when the value of an object changes and this causes the value of another property of an object to change. Static binding occurs when one property of an object is set directly, without changing another property.
- Dynamic binding also has more potential for errors because changing one property can cause unexpected changes in other properties and can even lead to crashing your app if something went wrong during execution.

Message Passing:

- When an object wants to send some information to another object, it can simply do so by calling the appropriate method on that other object.
- This allows objects to communicate with each other and build complex chains of objects together.
- Each object has a specific "address" that other objects can use to send them messages.
- When an object receives a message, it simply reads the message and does whatever it needs to do base on the message.

- In this way, the entire program can be made up of multiple objects communicating with each other. With message passing, it becomes much easier for programs to become more complex and for objects to interact with each other in interesting ways.