

In object-oriented programming (OOP), instances of a class are also referred to as objects. Objects are created based on a class definition, and each object represents a unique instance of that class.

## 1. Class Definition:

- A class is a blueprint or template that defines the structure and behavior of objects.
- It encapsulates data (attributes) and functions (methods) that operate on that data.

C++ 

```
// Class definition
class MyClass {
public:
    // Member function to set the value of the member variable
    void setValue(int val) {
        myValue = val;
    }

    // Member function to get the value of the member variable
    int getValue() {
        return myValue;
    }

private:
    // Private member variable
    int myValue;
};
```

## 2. Creating Instances (Objects):

- Objects are instances of a class. They represent specific entities based on the class blueprint.
- You create an object by declaring a variable of the class type.

C++



```
// Creating instances (objects) of MyClass
MyClass obj1; // obj1 is an object of MyClass
MyClass obj2; // obj2 is another object of MyClass
```

## 3. Manipulating Objects:

- Objects have their own set of member variables and can be manipulated using member functions.

C++



```
// Setting values for obj1 and obj2
obj1.setValue(42);
obj2.setValue(99);

// Retrieving values from obj1 and obj2
int value1 = obj1.getValue();
int value2 = obj2.getValue();
```

## 4. Independence of Objects:

- Each object created from the same class is independent of others.
- Changes to one object do not affect the state of other objects.

C++



```
// obj1 and obj2 have independent states
std::cout << "Value of obj1: " << value1 << std::endl;
std::cout << "Value of obj2: " << value2 << std::endl;
```

### Related posts:

1. Abstraction and encapsulation
2. Object Oriented Programming & Methodolog Viva Voce
3. How to install compiler for code blocks
4. Object Oriented Programming
5. Differences between Procedural and Object Oriented Programming
6. Features of Object Oriented Paradigm
7. Inheritance in Object Oriented Programming
8. Object Oriented Programming
9. Introduction to Object Oriented Thinking & Object Oriented Programming
10. Difference Between Object-Oriented Programming (OOP) and Procedural Programming
11. features of Object oriented paradigm
12. Merits and demerits of Object Oriented methodology
13. Concept of Objects: State, Behavior & Identity of an object
14. Access modifiers
15. Static members of a Class

- 16. Message Passing in OOP
- 17. Construction and destruction of Objects