OTZ

OPENTECHZ

A dream to realize your dream

OOP

Opentechz Pvt Ltd .
By Parthasarathi Swain

What is OOP? (OOP = Object-Oriented Programming)

Definition : OOP is a programming style where everything is represented as an **object**.

What is an object?

An object is a real-world entity (like a car, student, or bank account) that has:

Why OOP?

- Helps organize code around real-life things
- Makes code **modular** – each class handles a specific part
- Makes code **reusable** – write once, use again
- Makes code **maintainable** – easier to debug and update

Popular OOP Languages:

C++ ,Java ,Python

Features of Object-Oriented Programming (OOP)

- Class
- Object
- Encapsulation
- Abstraction
- Inheritance
- Polymorphism

What is class ?

A class is a user defined data type or blue print or specification or logical constructor of an object.

- **A class is user defined of an object because using class we can store multiple object.**
- **A class is called blue print of an object because using class we can create multiple objects of same type.**
- **A class is called specification of an object because it specify what an object contains.**
- **A class is also called logical constructor of an object because it constructs object logically(design) an object.**

What is an Object ?

A object is real world things which is an instance of a class .

- **State:** represent data of an object.
- **Behaviour:** represent the behaviour (functionality) of an object such as deposit, withdraw, sleep, fooding....etc
- **Identity:** An object indentity is typically implemented a via unique ID.

class object relation

```
#include <iostream>
using namespace std;

class Dog {
  public:
    string name;
    string breed;
    int age;

    void sleep() {
      cout << name << " is sleeping." << endl;
    }
};

int main() {
      Dog dog1;
            dog1.name = "Tommy";
            dog1.breed = "Labrador";
            dog1.age = 3;

            dog1.sleep();

  return 0;
}
```

OTZ OPENTECHZ
A dream to realize your dream

Members of a Class

- **Data Members (Variables / Attributes)**
- **Member Functions (Methods)**
- **Access Specifiers** (public, private, protected)
- **Constructors**
- **Destructors**
- **Static Members**

Access Specifiers in C++

Access specifiers control who can access class members (variables and functions).

public

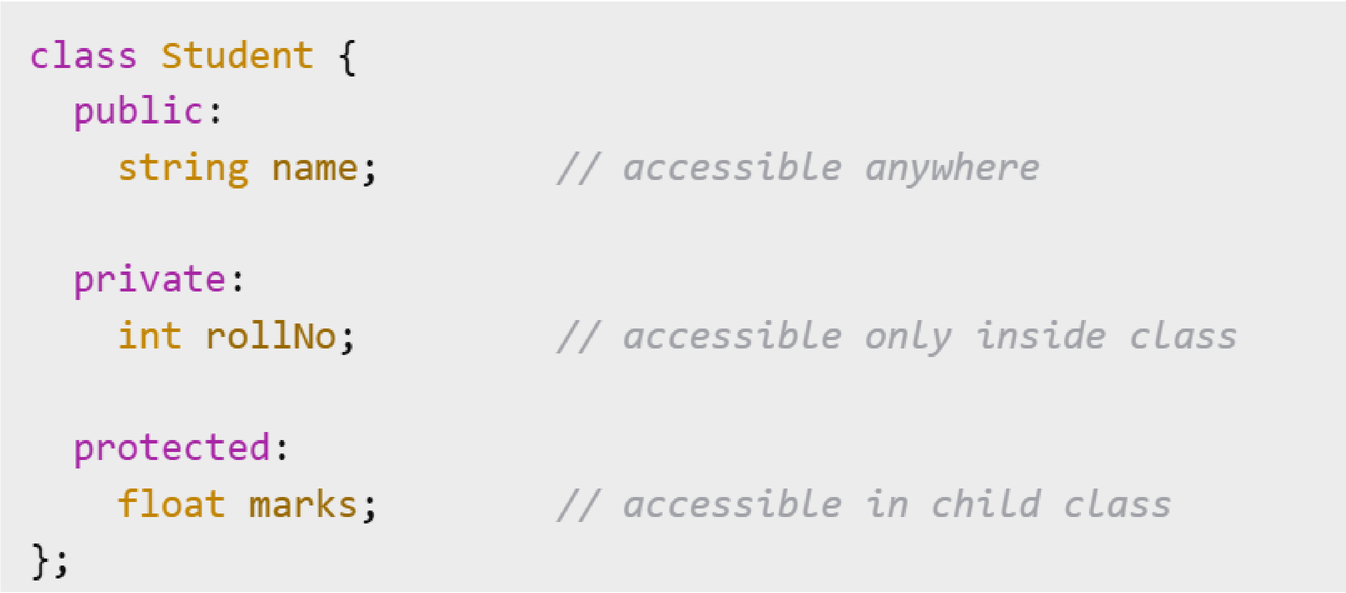
- Members are accessible from anywhere in the program.

private

- Members are accessible only inside the class.
- Not accessible from outside or from derived classes.

protected

- Members are accessible inside the class and in derived (child) classes.
- Not accessible from outside the class.



```
class Student {
  public:
    string name;          // accessible anywhere

  private:
    int rollNo;           // accessible only inside class

  protected:
    float marks;          // accessible in child class
};
```

OTZ

OPENTECHZ

A dream to realize your dream

Thank You