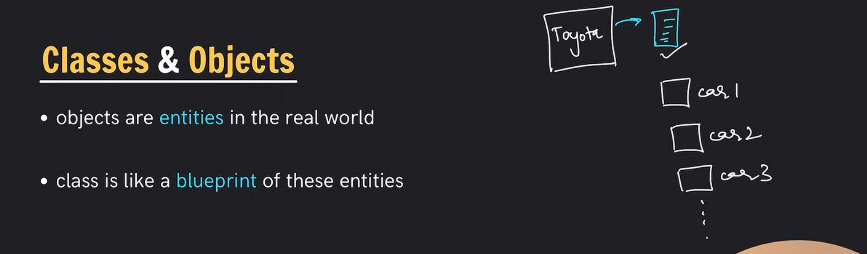
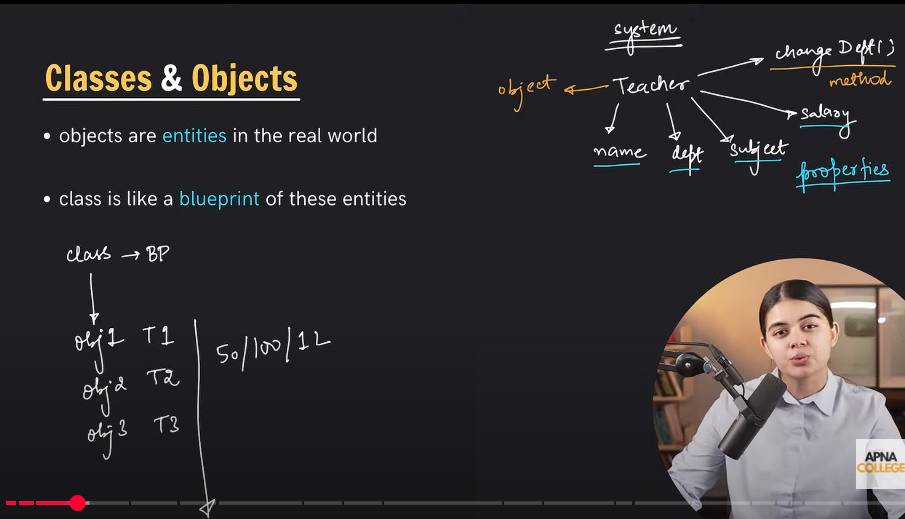
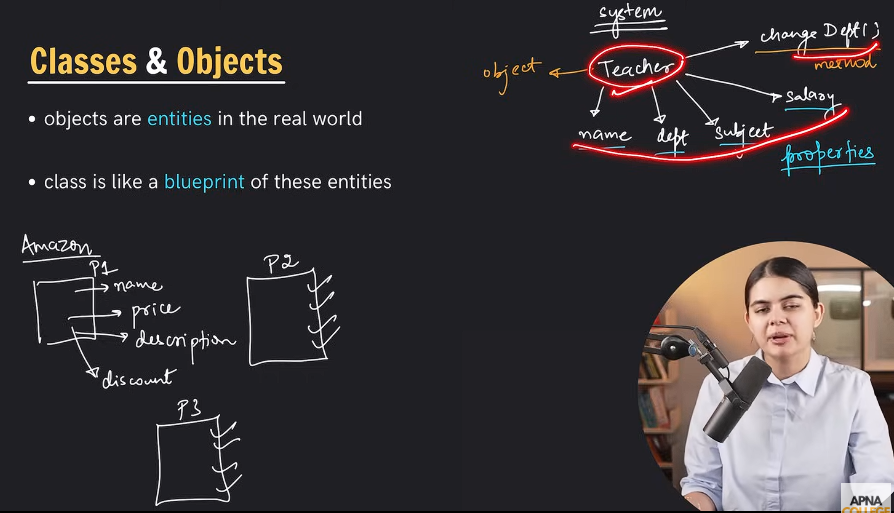
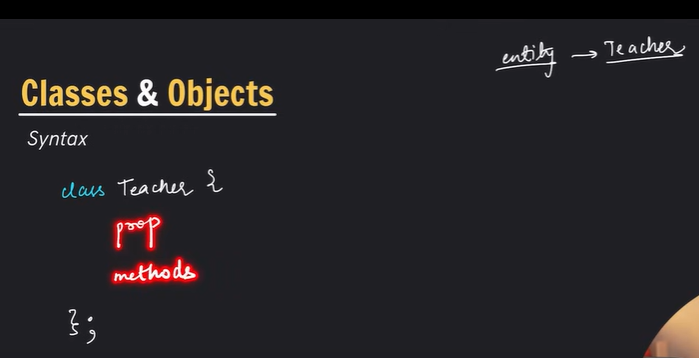
C++

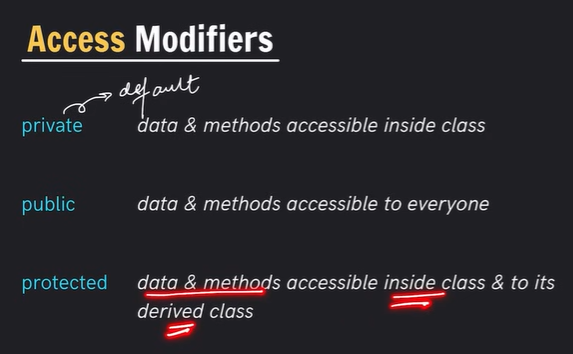
Object Oriented Programming











oop code:

#include<iostream>

using namespace std;

class Teacher

{

    private :

    double salary;

    public :

    //properties / attributes

    string name;

    string dept;

    string subject;

    //methods / member functions

    void changeDept(string newDept)

    {

        dept = newDept;

    }

};

int main()

{

    Teacher t1;

    t1.name = "rangan";

    t1.subject = "C++";

    t1.dept = "Computer Science";

    cout << "name : " << t1.name << endl;

    cout << "subject : " << t1.subject << endl;

    cout << "dept : " << t1.dept << endl;

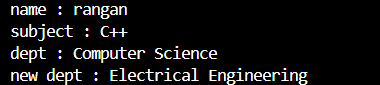
    t1.changeDept("Electrical Engineering");

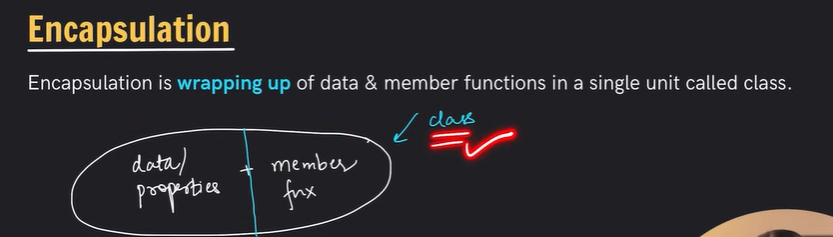
    cout << "new dept : " << t1.dept << endl;

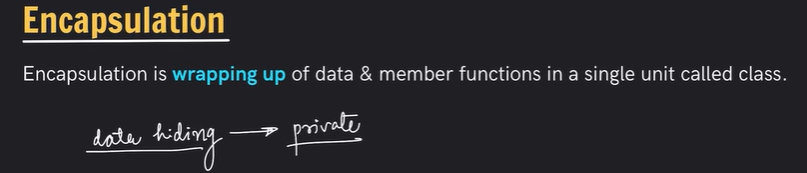
    return 0;

}

Result:







Getter, setter , printing private properties code:

#include<iostream>

using namespace std;

class Teacher

{

    private :

    double salary; // data hiding

    public :

    //properties / attributes

    string name;

    string dept;

    string subject;

    //methods / member functions

    void changeDept(string newDept)

    {

        dept = newDept;

    }

    //private er moddher properties er access pete setter getter use korte hobe

    void setSalary(double s) //setter

    {

        salary = s;

    }

    double getSalary() //getter

    {

        return salary;

    }

};

int main()

{

    Teacher t1;

    t1.name = "rangan";

    t1.subject = "C++";

    t1.dept = "Computer Science";

    t1.setSalary(25000);

    cout << "name : " << t1.name << endl;

    cout << "subject : " << t1.subject << endl;

    cout << "dept : " << t1.dept << endl;

    cout << "salary : " << t1.getSalary() << endl;

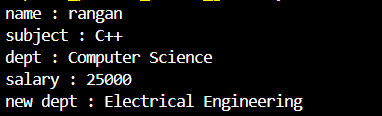
    t1.changeDept("Electrical Engineering");

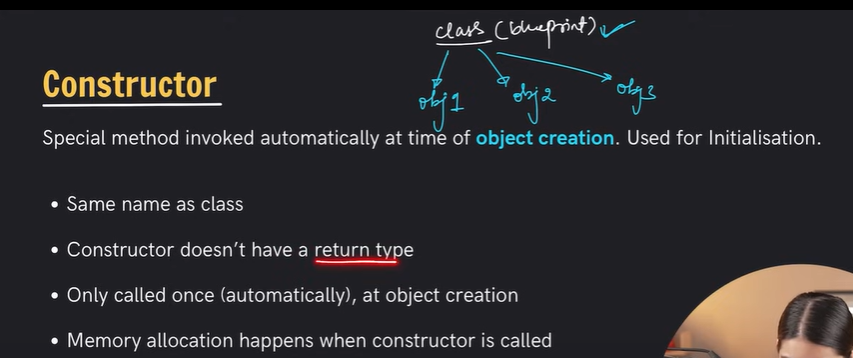
    cout << "new dept : " << t1.dept << endl;

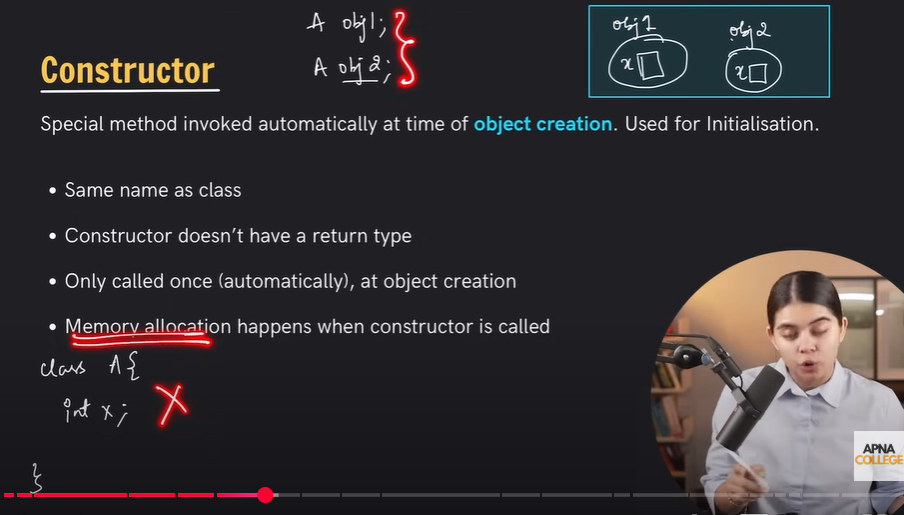
    return 0;

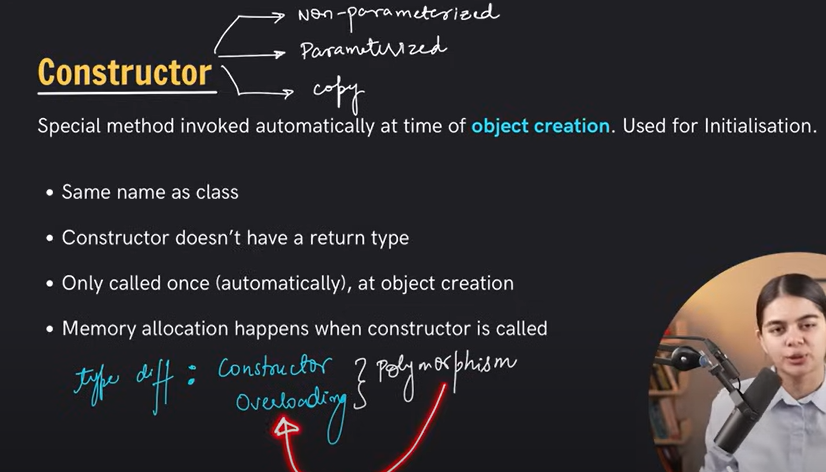
}

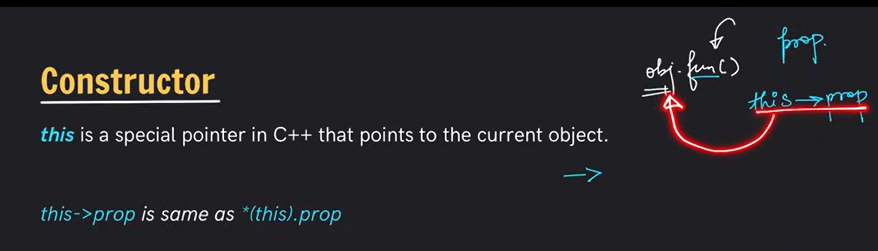
Result:











Constructor code**:**

#include<iostream>

using namespace std;

class Teacher

{

    private :

    double salary; // data hiding

    public :

    //properties / attributes

    string name;

    string dept;

    string subject;

    //constructor

    //constructor ke sobshomoy public declare korte hobe

    Teacher()//non-parameterized constructor

    {

        cout << "Hi! I am constructor-1\n";

        dept = "Computer Science";

    }

    Teacher(string n,string dept, string subject, double sal)//parameterized constructor

    {

        cout << "Hi! I am constructor-2\n";

        name = n;

        this -> dept = dept; //same name paprameter dile this-> use kote hobe

        this -> subject = subject; //this-> diye object er name ke inicate kore

        salary = sal;

    }

    //methods / member functions

    void changeDept(string newDept)

    {

        dept = newDept;

    }

    //private er moddher properties er access pete setter getter use korte hobe

    void setSalary(double s) //setter

    {

        salary = s;

    }

    double getSalary() //getter

    {

        return salary;

    }

    void getInfo()

    {

        cout << "name : " << name << endl;

        cout << "dept : " << dept << endl;

        cout << "subject : " << subject << endl;

        cout << "salary : " << getSalary() << endl;

    }

};

int main()

{

    Teacher t1; //constructor call

    t1.name = "rangan";

    t1.subject = "C++";

    t1.setSalary(25000);

    Teacher t2("adnan", "EEE", "Physics", 20000); //constructor call

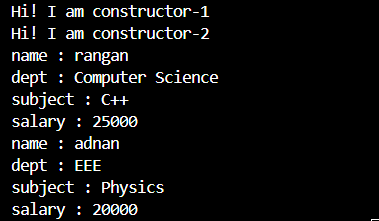
    t1.getInfo();

    t2.getInfo();

    return 0;

}

**Result:**



Copy constructor:

#include<iostream>

using namespace std;

class Teacher

{

    private :

    double salary; // data hiding

    public :

    //properties / attributes

    string name;

    string dept;

    string subject;

    Teacher(string n,string dept, string subject, double sal)//parameterized constructor

    {

        cout << "Hi! I am constructor-2\n";

        name = n;

        this -> dept = dept; //same name paprameter dile this-> use kote hobe

        this -> subject = subject; //this-> diye object er name ke inicate kore

        salary = sal;

    }

    //copy constructor

    Teacher(Teacher &orgObj) //pass by reference

    {

        cout << "I'm custom copy constructor....\n";

        this -> name = orgObj.name;

        this -> dept = orgObj.dept;

        this -> subject = orgObj.subject;

        this -> salary = orgObj.salary;

    }

    //methods / member functions

    void changeDept(string newDept)

    {

        dept = newDept;

    }

    //private er moddher properties er access pete setter getter use korte hobe

    void setSalary(double s) //setter

    {

        salary = s;

    }

    double getSalary() //getter

    {

        return salary;

    }

    void getInfo()

    {

        cout << "name : " << name << endl;

        cout << "dept : " << dept << endl;

        cout << "subject : " << subject << endl;

        cout << "salary : " << getSalary() << endl;

    }

};

int main()

{

    Teacher t1("adnan", "EEE", "Physics", 20000); //constructor call

    Teacher t2(t1);//custom copy constructor invoke

    //custom copy constructor create na korleo compiler nije theke default copy constructor create korbe

    //amader copy constructor create kora necessary na

    t2.getInfo();

    return 0;

}

Result:

