

```

#include <iostream>

using namespace std;

class Height
{
public:
    int feet, inch;

    Height()
    {
        feet = 0;
        inch = 0;
    }

    Height(int f, int i)
    {
        feet = f;
        inch = i;
    }

    // Overloading (+) operator to perform addition of
    // two distance object using binary operator
    Height operator+(Height& d2) // Call by reference
    {
        // Create an object to return
        Height h3;

        // Perform addition of feet and inches
        h3.feet = feet + d2.feet;
        h3.inch = inch + d2.inch;

        // Return the resulting object
        return h3;
    }

};

int main()
{

```

```

Height h1(3, 7);
Height h2(6, 1);
Height h3;
//Use overloaded operator
h3 = h1 + h2;
cout << "Sum of Feet & Inches: " << h3.feet << "" << h3.inch << endl;
return 0;
}

```

```

#include<iostream>
using namespace std;

```

```

class Point
{
private:
    int x, y;
public:
    Point(int x1, int y1) { x = x1; y = y1; }

    // Copy constructor
    Point(const Point &p2) {x = p2.x; y = p2.y; }

    int getX()          { return x; }
    int getY()          { return y; }
};

```

```

int main()
{ int t,y;
cout<<"Enter 2 int values\n";
cin>>t>>y;

    Point p1(t, y); // Normal constructor is called here

```

```

        Point p2=p1; // Copy constructor is called here

        // Let us access values assigned by constructors
        cout << "p1.x = " << p1.getX() << ", p1.y = " << p1.getY();
        cout << "\np2.x = " << p2.getX() << ", p2.y = " << p2.getY();

        return 0;
    }

```

```

// Cpp program to illustrate the
// concept of Constructors

#include <iostream>

using namespace std;

```

```

class cons{
public:
    int p, q;

    // Default Constructor
    cons();

    void read();
};

```

```

cons::cons()
{
    p = 78;
    q = 56;
}

```

```

void cons::read()
{

```

```

int d,k;

cout<<"Enter d & k int values\n";

cin>>d>>k;

cout<<"d"<<d<<"k"<<k;

cout<<"Enter p, q values\n";

cin>>p>>q;

cout<<"p"<<p<<"q"<<q;

```

```

}

int main()
{
    // Default constructor called automatically
    // when the object is created

    cons c;

    cout << "p: " << c.p << endl << "q: " << c.q;

    c.read();

    return 0;
}

```

```

#include <iostream>

#include<cstring>

#include<bits/stdc++.h>

using namespace std;

class DemoDC {

    public:

    int id,marks,i;

    char name[20];

    DemoDC() {

```

```

    id=0;

    marks=0;

    /*for(i=0;i<20;i++)
    {
        name[i]=" ";
    }*/
}

//address=" ";

void create()
{
    cin.ignore();
    cout<<"Enter name";

    cin.getline(name,20);
    cout<<"Enter id";
    cin>>id;
    cout<<"Enter marks";
    cin>>marks;

    //cout<<"Enter address";
    //cin>>address;

}

void display()
{

    cout<<"id = "<< id <<endl;
    cout<<"marks = "<< marks <<endl;
    cout<<"name = "<< name <<endl;

```

```
//cout<<"address= "<< address<<endl;

}

};

int main()
{
    DemoDC obj;
    DemoDC obj1[5];
    int n;
    cout<<"Enter limit";
    cin>>n;
    for(int i=0;i<n;i++)
    {
        obj.create();

        obj.display();
    }

    for(int i=0;i<n;i++)
    {
        obj1[i].create();

        obj1[i].display();
    }
    for(int i=0;i<n;i++)
    {
        cout<<"Enter name\n";
        cin>>obj1[i].name;
        cout<<"Enter marks";
        cin>>obj1[i].marks;
    }
}
```

```
    return 0;
}

#include<iostream>
using namespace std;
class Emp
{
public:
    long long int empid,salary,mobno;
    string name,address;
    Emp()
    {
        empid=0;
        mobno=0;
        salary=0;
        name=" ";
        address=" ";
    }
    void create()
    {
        cout<<"Enter Empid";
        cin>>empid;
        cout<<"Enter mobno";
        cin>>mobno;
        cout<<"Enter Salary";
        cin>>salary;
        cout<<"Enter name";
        cin>>name;
        cout<<"Enter Address";
        cin>>address;
    }
}
```

```

void display()
{cout<<"Empid"<<empid<<"\n";

    cout<<"Mobno"<<mobno<<"\n";

    cout<<"Salary"<<salary<<"\n";

    cout<<"Name"<<name<<"\n";

    cout<<"Address"<<address<<"\n";

}

};

int main()
{

    Emp e;

    char ch;

do
{

    e.create();

    e.display();

    cout<<"Do you want to continue";

cin>>ch;

}while(ch=='y' || ch=='Y');

    return 0;

}

```

```

// CPP program to illustrate
// parameterized constructors

#include <iostream>

using namespace std;

class Point {

```



private:

int x, y;

public:

// Parameterized Constructor

Point(int x1, int y1)

{

x = x1;

y = y1;

}

int getX()

{

return x;

}

int getY()

{

return y;

}

};

int main()

{

// Constructor called

Point p1(10, 15);

int m,n;

cout<<"Enter int values for m,n \n";

cin>>m>>n;

// Access values assigned by constructor

```

        cout << "p1.x = " << p1.getX() << ", p1.y = " << p1.getY();
Point p2(m,n);

        cout << "p2.x = " << p2.getX() << ", p2.y = " << p2.getY();


        return 0;
}

#include <iostream>
#include<iomanip>
using namespace std;
class Height {
public:
// Member Objects
int feet, inch;
// Constructor to initialize the object's value
Height(int f, int i)
{
feet = f;
inch = i;
}
// Overloading(-) operator to perform decrement
// operation of Height object
void operator-()
{
feet--;
inch--;
cout << "Feet & Inches after decrement: " << feet << " ' " << inch << endl;
}
};

int main()
{
//Declare and Initialize the constructor of class Height

```

```
Height h1(6, 2);  
//Use (-) unary operator by single operand  
-h1;  
return 0;  
}
```