

```

#include <iostream.h>
#include <conio.h>
class Box
{
    int l,b,h;
public:
    Box();
    Box(int);
    Box(int,int,int);
    Box(Box&);
    void show();
};

Box::Box()
{
    cout<<"enter l,b,h:";
    cin>>l>>b>>h;
}

Box::Box(int i,int j,int k)
{
    l=i;
    b=j;
    h=k;
}

Box::Box(Box &P)
{
    l=P.l;
    b=P.b;
    h=P.h;
}

void Box::show()
{
    cout<<l<<" "<<b<<" "<<h<<endl;
}

int main()
{
    clrscr();
    Box B1;
    Box B2(10);
    Box B3(6,7,9);
    Box B4(B1);
    B1.show();
    B2.show();
    B3.show();
    B4.show();
    getch();
    return 0;
}

```

Handwritten diagrams and annotations:

- Diagram of B1: l=10, b=15, h=20. An arrow points from B1 to the **Box B4(B1);** line in the code.
- Diagram of B2: l=10, b=10, h=10. An arrow points from B2 to the **B2.show();** line in the code.
- Diagram of B3: l=6, b=7, h=9.
- Diagram of B4: l=10, b=15, h=20.
- Handwritten notes: "B1", "P", "B2", "B3", "B4" are written near their respective diagrams.

```

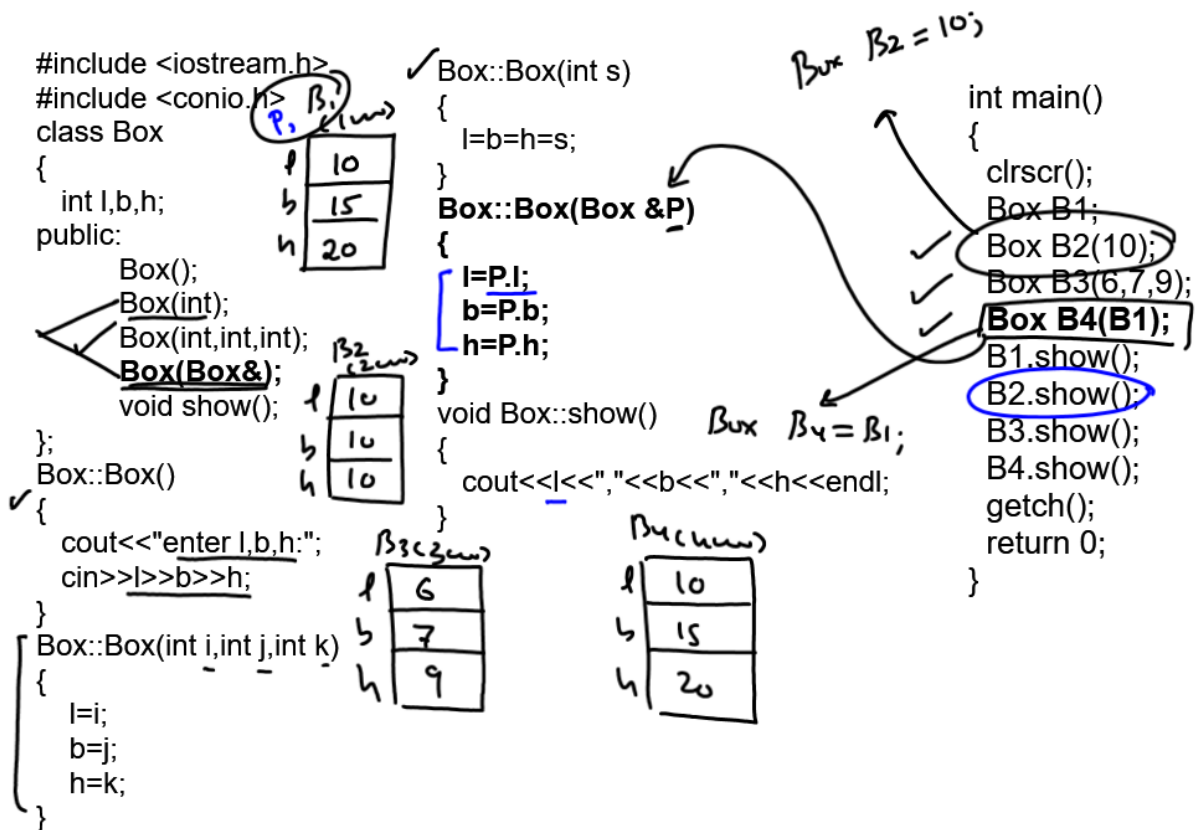
Box::Box(Box &P)
{
    l=P.l;
    b=P.b;
    h=P.h;
}

```

```

int main()
{
    clrscr();
    Box B1;
    Box B2(10);
    Box B3(6,7,9);
    Box B4(B1);
    Box B5(B3);
    B1.show();
    B2.show();
    B3.show();
    B4.show();
    getch();
    return 0;
}

```



`Box B2(10);`
 OK
`Box b2 = 10;`

Same

`Box B4(B1);`
 OK
`Box B4 = B1;`

Same

Box B₄ = B₁ → Copy Const Call

v/s

Box B₄;
B₄ = B₁;

Non-Param Const Call

Copying done by assignment

```
#include <iostream.h>
#include <conio.h>
class Box
{
    int l,b,h;
public:
    Box();
    Box(int);
    Box(int,int,int);
    Box(Box&);
    void show();
};

Box::Box()
{
    cout<<"enter l,b,h:";
    cin>>l>>b>>h;
}

Box::Box(int i,int j,int k)
{
    l=i;
    b=j;
    h=k;
}

Box::Box(Box &P)
{
    l=P.l;
    b=P.b;
    h=P.h;
}

void Box::show()
{
    cout<<l<<" "<<b<<" "<<h<<endl;
}

Box B2 = 10;

int main()
{
    clrscr();
    Box B1;
    Box B2(10);
    Box B3(6,7,9);
    Box B4(B1);
    B1.show();
    B2.show();
    B3.show();
    B4.show();
    getch();
    return 0;
}
```

Box B₁ (l=10, b=15, h=20)

Box B₂ (l=10, b=10, h=10)

Box B₃ (l=6, b=7, h=9)

Box B₄ (l=10, b=15, h=20)

How many constructors this class has ?

```
class Box
{
    int l,b,h;
public:
    void get();
    void show();
};
```

Answer: 2 , both are given by compiler: default cont and default copy const

How many constructors this class has ?

```
class Box
{
    int l,b,h;
public:
    Box();
    void show();
};
```

Answer: 2 , ONE is programmer's const caLLED NON PARAMETRIZED CONSTRUCTOR and dfault copy const

How many constructors this class has ?

```
class Box
{
    int l,b,h;
public:
    Box();
    Box(int);
    void show();
};
```

Answer: 3 , TWO by programmer's and one by compiler default copy const

How many constructors this class has ?

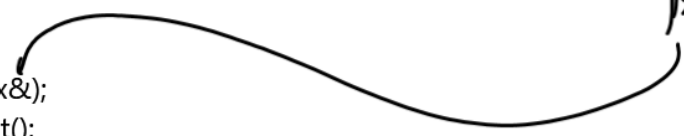
```
class Box
{
    int l,b,h;
public:
    Box();
    Box(int);
    Box(Box&);
    void show();
};
```

Answer: 3 , All by programmer

How many constructors this class has ?

```
class Box
{
    int l,b,h;
public:
    Box(Box&);
    void get();
    void show();
};
```

~~X~~ Box B1;
B1.get();
Box B2(B1);



Answer: 1: It is programmer defined copy constructor