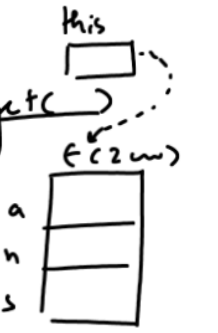
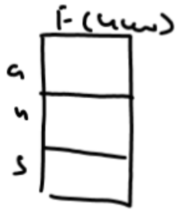
The "this" Pointer

```
class Emp
{
    int age;
    char name[20];
    float sal;
public:
    void get( );
    void show( );
};
```

```
void Emp::get( )
{
    =
}
```



```
void Emp::show( )
{
    =
}
```



```
int main( )
{
✓ Emp E, F;
  E.get( );
  E.show( );
  F.get( );
  F.show( );
  ret 0;
}
```

What is "this"?
============

"this" is a special pointer available inside every member function of a class except it static member function. Whenever we call any non-static member function ofthe class then the C++ compiler automatically passes the address of the calling object to that member function. Within the member function, compiler receives this address in a special pointer and this pointer is called "this" pointer.

So, we can say that in C++ every non-static member function always knows the address of its calling object through its "this" pointer.

Moreover since "this" is a pointer, like other pointers it occupies 8 byte of size or 4 bytes of size or 2 bytes of size as per compiler's architechture and the data type of "this" is always class name.

```
#include <iostream>
using namespace std;
class Emp
{
    int age;
    char name[20];
    float sal;
public:
    void get();
    void show();
};
void Emp::get()
{
    cout<<"My calling object's address is:"<<(unsigned long long int)this<<endl;
    cout<<"Enter age ,name and sal:";
    cin>>age>>name>>sal;
}
void Emp::show()
{
    cout<<"My calling object's address is:"<<(unsigned long long int)this<<endl;
    cout<<age<<","<<name<<","<<sal<<endl;
}
```

```
int main()
{
    Emp E,F;
    cout<<"Address of E:"<<(unsigned long long int)&E<<endl;
    cout<<"Address of F:"<<(unsigned long long int)&F<<endl;
    E.get();
    F.get();
    E.show();
    F.show();
    return 0;
}
```

Accessing Object's Values Using "this"

```
#include <iostream>
using namespace std;
class Emp
{
    int age;
    char name[20];
    float sal;
public:
    void get();
    void show();
};
void Emp::get()
{

    cout<<"Enter age ,name and sal:";
    cin>>this->age>>this->name>>this->sal;
}
void Emp::show()
{

    cout<<this->age<<","<<this->name<<","<<this->sal<<endl;
}
```

```
int main()
{
    Emp E,F;

    E.get();
    F.get();
    E.show();
    F.show();
    return 0;
}
```

Benefits of using "this"
================
By using "this" pointer we can get 3 benefits.

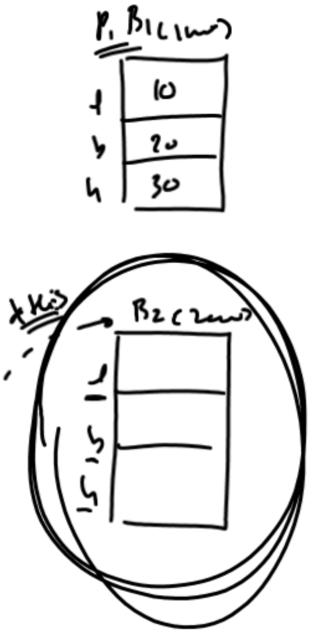
1. We can resolve the overlapping of class members done by local variables of the same name inside the member function.

2. By using "this" we can reduce the body of copy constructor to one single line.

3. By using "this" we can reduce number of statements in the body of overloaded operator functions.

```
#include <iostream>
using namespace std;
class Box
{
    int l,b,h;
public:
    Box(int,int,int);
    Box(Box&);
    void show();
};
Box::Box(int l,int b,int h)
{  ✓
    this->l=l;
    this->b=b;
    this->h=h;
}
Box::Box(Box &P)
{
    *this=P;   this
}
```

→ For example: B2=B1;



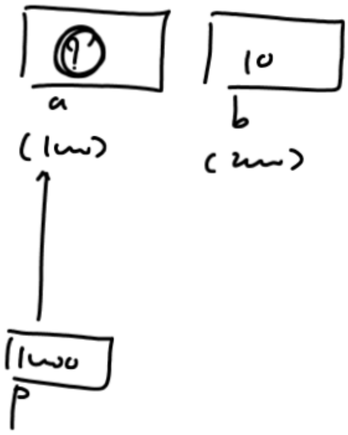
```
void Box::show()
{
cout<<l<<","<<b<<","<<h<<endl;
}
int main()
{
✓  Box B1(10,20,30);
✓  Box B2(B1);
   B1.show();
   B2.show();
   return 0;
}
```

```
int   a,  b= 10;

int  *p;

p= &a;

a = b;
 or
*p= b;
```


?
a
(1000)
10
b
(2000)
1000
p