/*

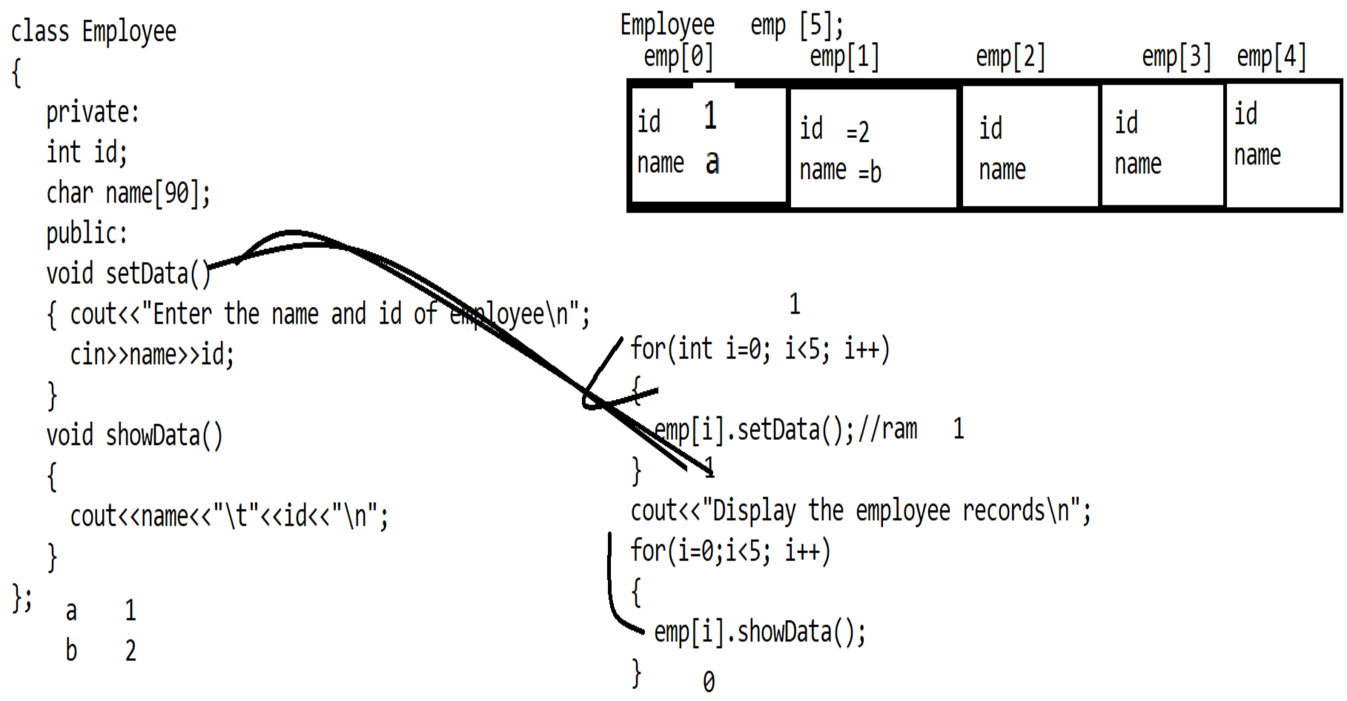
Array of objects in C++

Array of objects is used for store the more one object data in to the single name object called as array of objects.

How to Create the Array of objects in C++

Syntax: classname objectname[size];

Following Diagram Demonstrate the Array of objects



```
class Employee
{
   private:
   int id;
   char name[90];
   public:
   void setData()
   { cout<<"Enter the name and id of employee\n";
     cin>>name>>id;
   }
   void showData()
   {
     cout<<name<<"\t"<<id<<"\n";
   }
};  a   1
    b   2
```

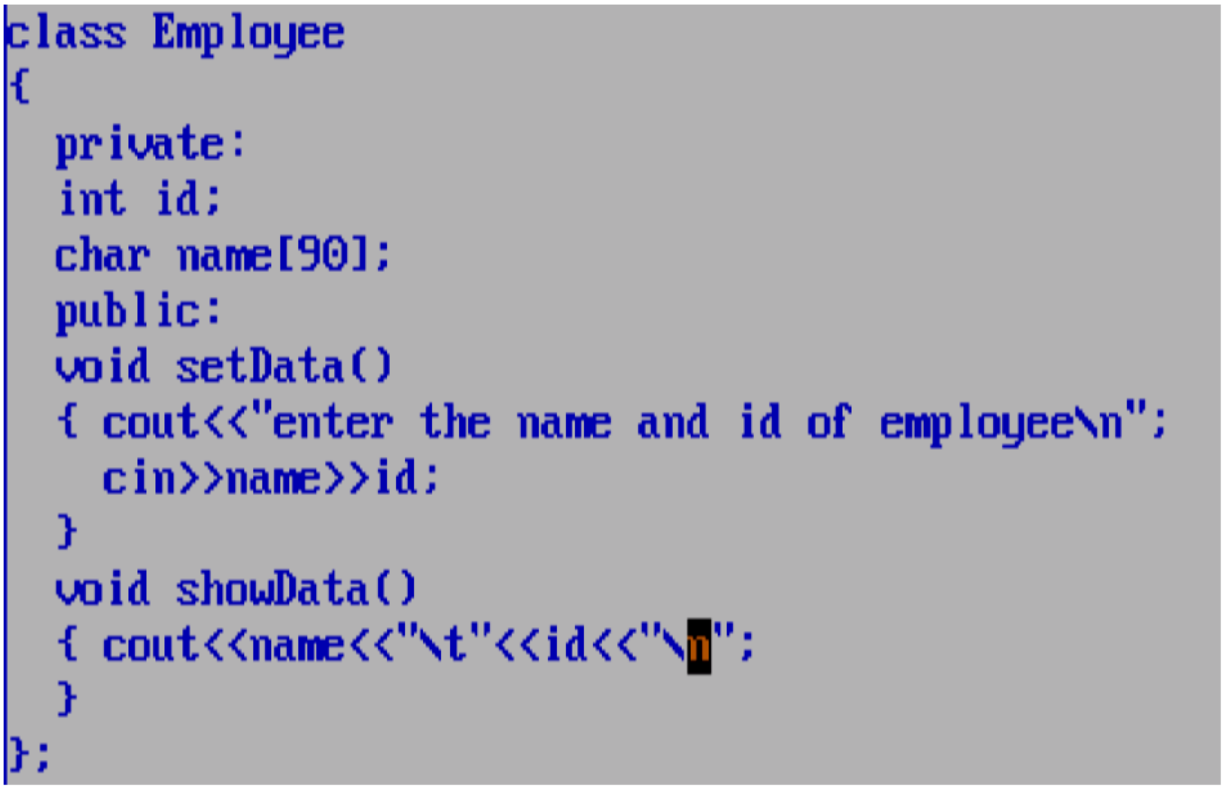
Employee   emp [5];

emp[0]	emp[1]	emp[2]	emp[3]	emp[4]

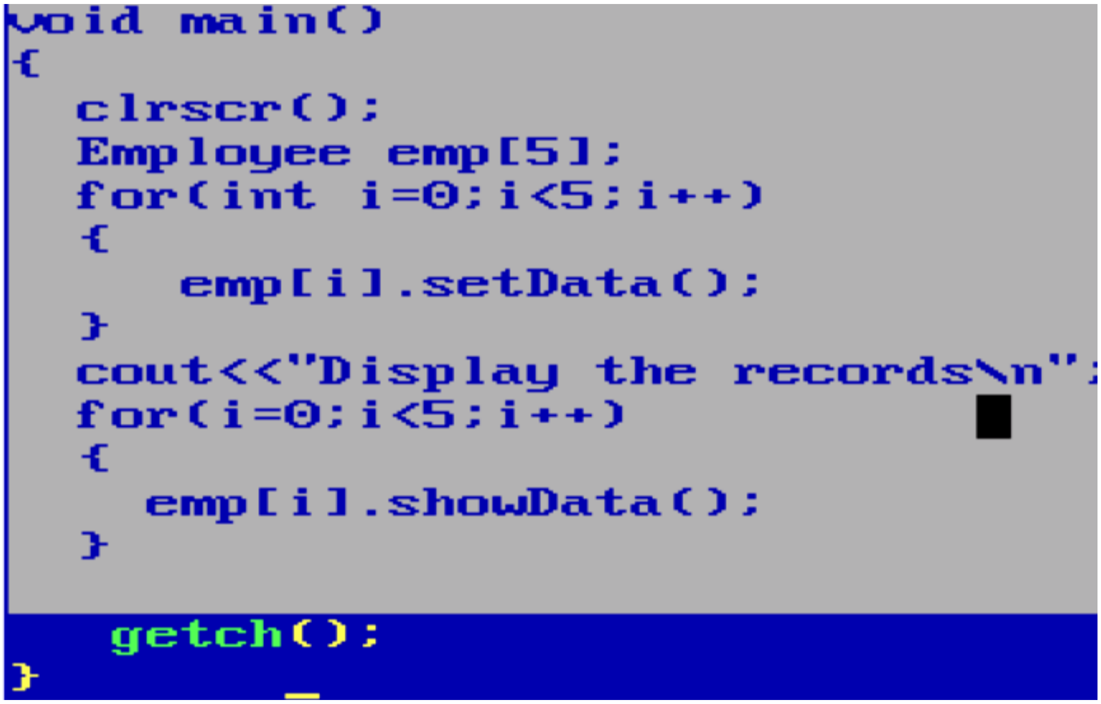
id 1
name a
	id =2
	name =b
		id
		name
			id
			name
				id
				name

```
                    1
for(int i=0; i<5; i++)
{
 emp[i].setData();//ram   1
}   1
cout<<"Display the employee records\n";
for(i=0;i<5; i++)
{
 emp[i].showData();
}   0
```

Following Example Demonstrate the Array of Objects

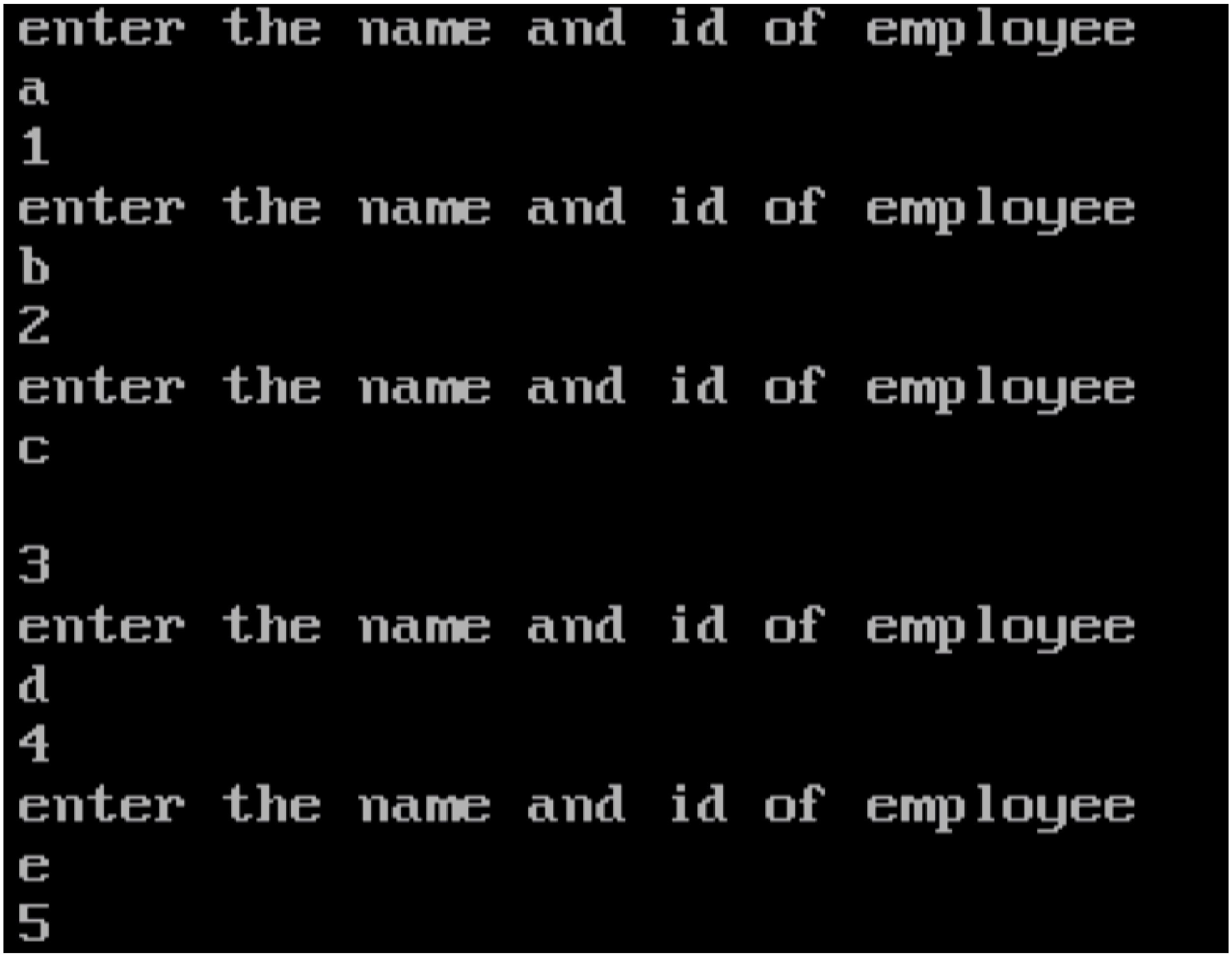


```
class Employee
{
  private:
  int id;
  char name[90];
  public:
  void setData()
  { cout<<"enter the name and id of employee\n";
    cin>>name>>id;
  }
  void showData()
  { cout<<name<<"\t"<<id<<"\n";
  }
};
```

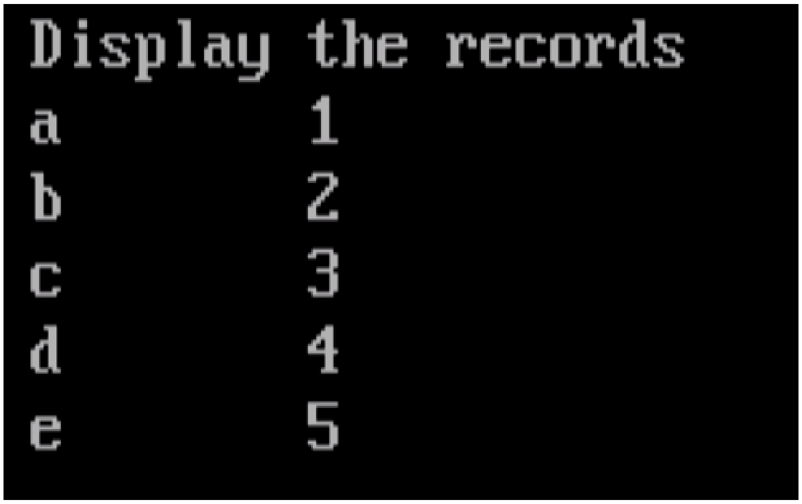


```
void main()
{
   clrscr();
   Employee emp[5];
   for(int i=0;i<5;i++)
   {
      emp[i].setData();
   }
   cout<<"Display the records\n";
   for(i=0;i<5;i++)
   {
     emp[i].showData();
   }

    getch();
}
```

Output



```
enter the name and id of employee
a
1
enter the name and id of employee
b
2
enter the name and id of employee
c

3
enter the name and id of employee
d
4
enter the name and id of employee
e
5
```



```
Display the records
a       1
b       2
c       3
d       4
e       5
```

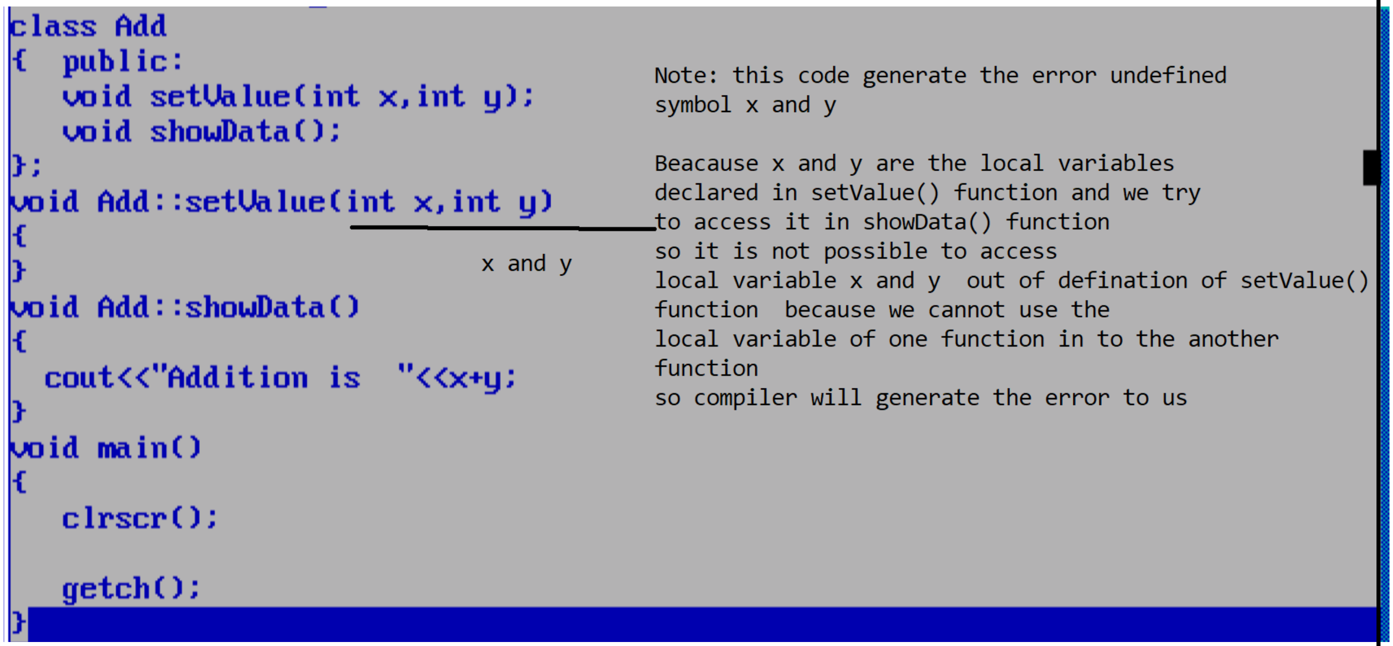
Local Variables

Local variables means if we define the variable within function called as local variable.

```
class ABC
{
   public:
   void setValue(int x,int y)
   {                          x and y are the local variable
   }
};
```

If we want to work with local variable we have the some important points given below

1) local variable cannot access outside of his definition



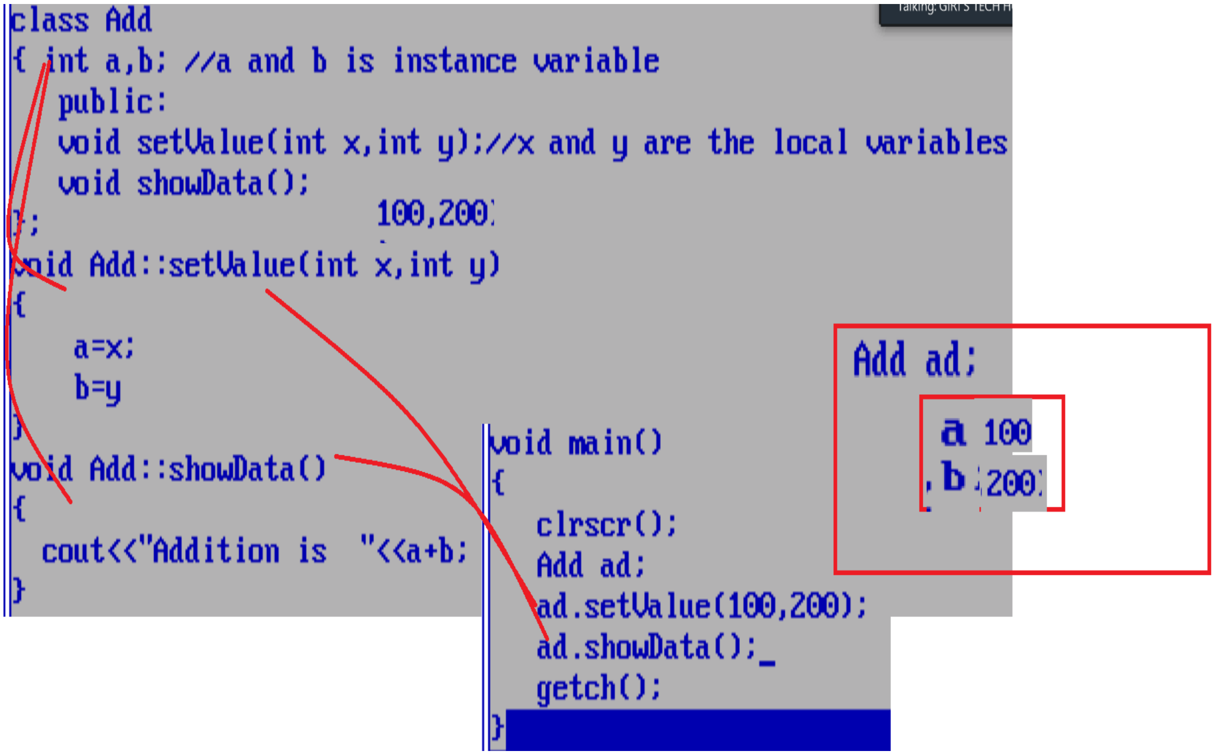
```
class Add
{  public:
   void setValue(int x,int y);
   void showData();
};
void Add::setValue(int x,int y)
{
}                                    x and y
void Add::showData()
{
  cout<<"Addition is  "<<x+y;
}
void main()
{
   clrscr();

   getch();
}
```

Note: this code generate the error undefined
symbol x and y

Beacause x and y are the local variables
declared in setValue() function and we try
to access it in showData() function
so it is not possible to access
local variable x and y out of defination of setValue()
function because we cannot use the
local variable of one function in to the another
function
so compiler will generate the error to us

If we want to solve this problem then we have to declare the instance variable in class and store the values of local variable in instance variable and instance variable can access in all functions defined under the class.

Following Example demonstrate the above code statement meaning



```
class Add
{ int a,b; //a and b is instance variable
   public:
   void setValue(int x,int y);//x and y are the local variables
   void showData();
};
void Add::setValue(int x,int y)
{
    a=x;
    b=y
}
void Add::showData()
{
  cout<<"Addition is  "<<a+b;
}
```

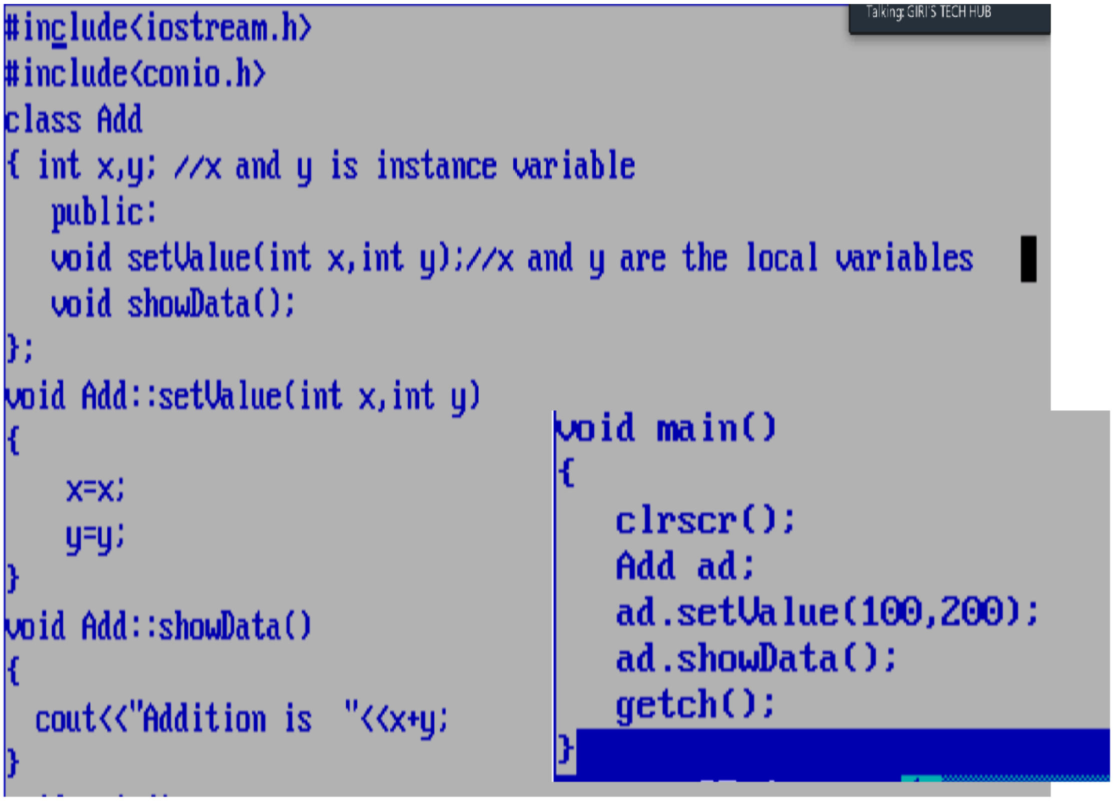
```
void main()
{
   clrscr();
   Add ad;
   ad.setValue(100,200);
   ad.showData();_
   getch();
}
```

In above example we have the x and y is local variable and we copy the content of x and y in instance variable name as a and b and a

and b instance variable can access in all functions in class.

2) If instance variable name and local variable is same then instance variable never access in block where local variable name is same

Following Example Shows the meaning of above statement



```
#include<iostream.h>
#include<conio.h>
class Add
{ int x,y; //x and y is instance variable
   public:
   void setValue(int x,int y);//x and y are the local variables
   void showData();
};
void Add::setValue(int x,int y)
{
    x=x;
    y=y;
}
void Add::showData()
{
  cout<<"Addition is  "<<x+y;
}
```

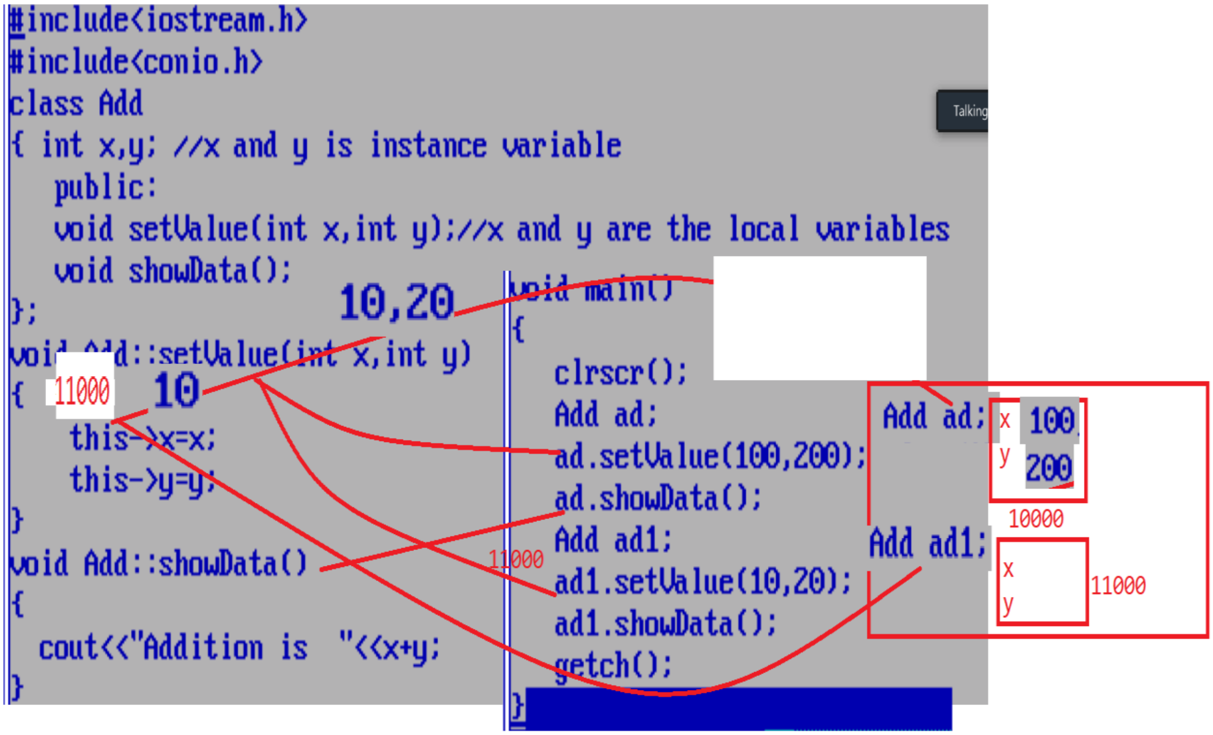
```
void main()
{
   clrscr();
   Add ad;
   ad.setValue(100,200);
   ad.showData();
   getch();
}
```

If we think about above code we have the output is Addition is garbage or zero. Because we give the name of local variable and name of instance variable is same If we want to avoid this problem in C++ we have this reference.

this reference

this is internal pointer present in every class which is used to point current running object in memory.

Normally t his pointer reference by people when they have same name instance variable and same name local variable.



```
#include<iostream.h>
#include<conio.h>
class Add
{ int x,y; //x and y is instance variable
   public:
   void setValue(int x,int y);//x and y are the local variables
   void showData();
};
void Add::setValue(int x,int y)
{
    this->x=x;
    this->y=y;
}
void Add::showData()
{
  cout<<"Addition is  "<<x+y;
}
```

```
void main()
{
   clrscr();
   Add ad;
   ad.setValue(100,200);
   ad.showData();
   Add ad1;
   ad1.setValue(10,20);
   ad1.showData();
   getch();
}
```