new keyword:

new keyword is used for create the dynamic or allocate dynamic memory in CPP.

How to create Dynamic Array by using new keyword

Syntax: datatype *variablename = new datatype[size];

```
#include<iostream.h>
#include<conio.h>

void main()
{    clrscr();
   int size;
   cout<<"Enter the size of array\n";
   cin>>size;
   int  *ptr = new int[size];
   cout<<"Enter the vaules in array\n";
   for(int i=0; i<size; i++)
   {
     cin>>ptr[i];
   }
   cout<<"Display the array values\n";
   for(i=0;i<size;i++)
   {
     cout<<ptr[i]<<"\n";
   }
      getch();
}
```

How to Create Dynamic object by using new keyword

Syntax: classname *ref =new classname();

Example

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

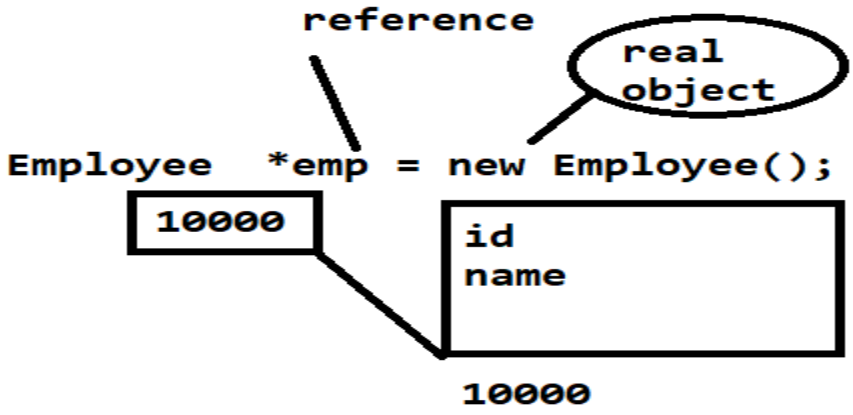
Example of Object

Employee *emp = new Employee ();

Here we can say emp is reference of Employee and new Employee () is real object.

Q. What is the diff between reference and object?

Reference is variable which hold the address of object and object is block of memory where class data.



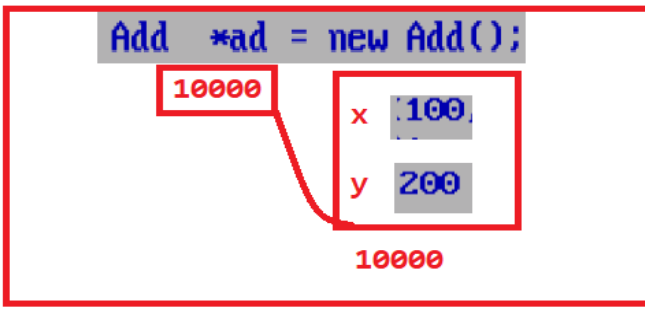
Q. Why we need to use the reference?

If we want to reuse the same object more than one time then we can use the reference.

```
#include<iostream.h>
#include<conio.h>
class Add
{
  int x,y;
  public:
  void setValue(int a,int b);
  void showAdd();
};
void Add::setValue(int a,int b)
{
  x=a;
  y=b;
}
void Add::showAdd()
{  cout<<"Addition is  "<<x+y<<"\n";
}
```

```
void main()
{
   clrscr();
   Add  *ad = new Add();
   ad->setValue(100,200);
   ad->showAdd();
   getch();
}
```

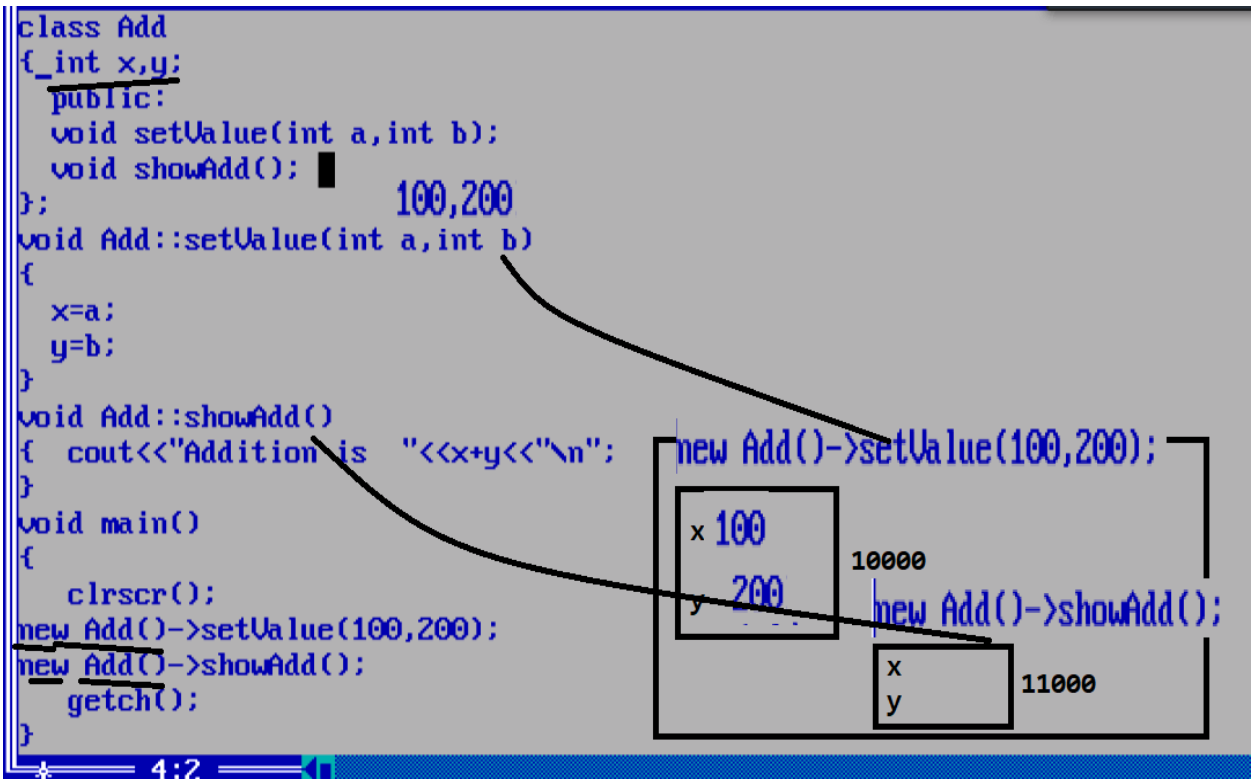
```
Addition is  300
```



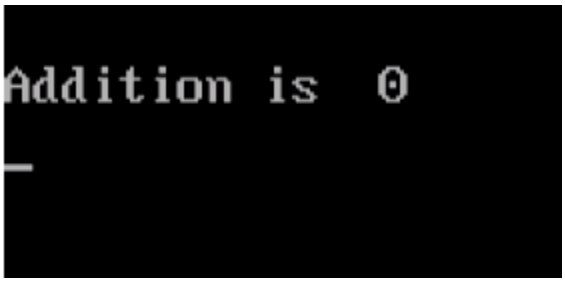
Q.what happens if we not use the reference with object?

If we not use the reference with object then internally every time new keyword create new object in memory means we have new block in memory every time so we cannot use the single object multiple time with multiple function with multiple places

If we want to reuse the same object multiple then we have to use the reference with object.



Output



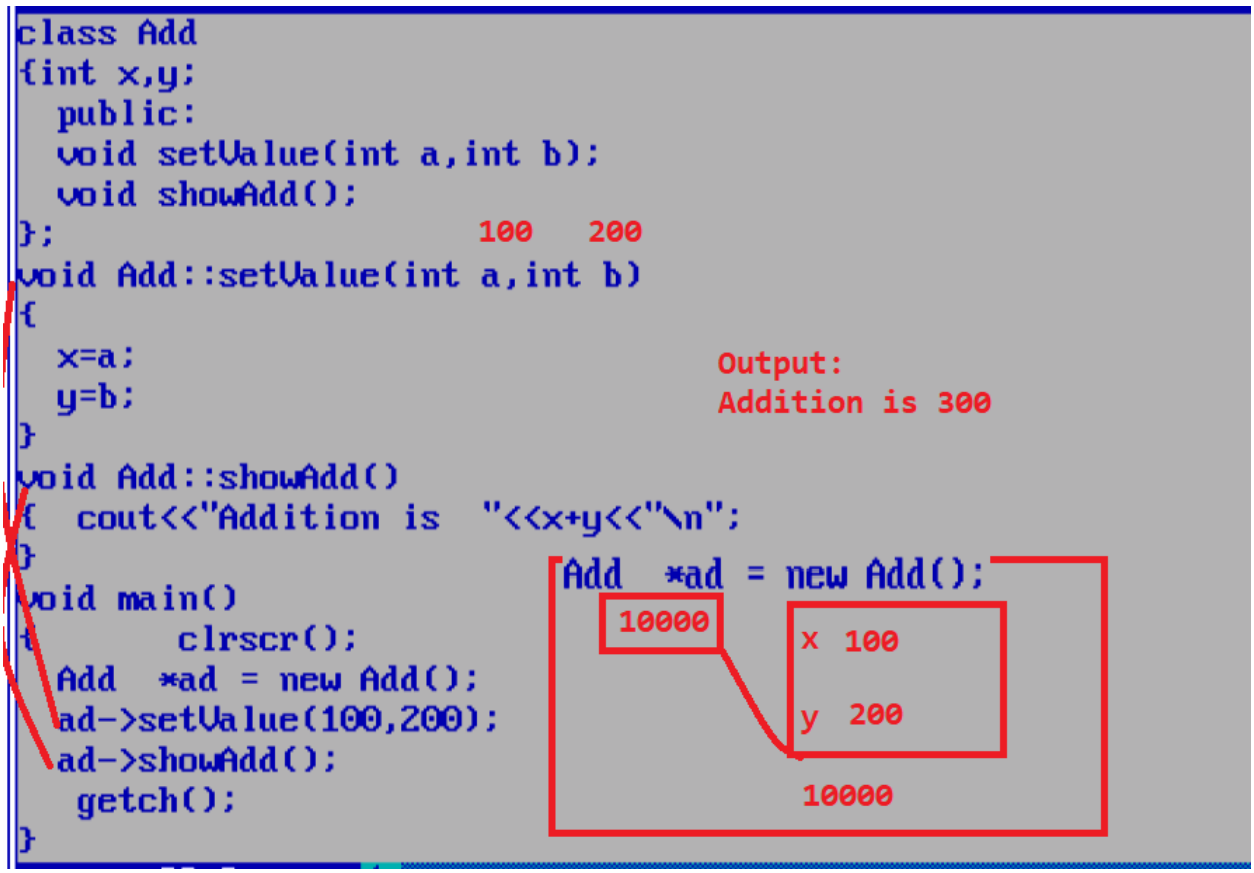
If we think about the above output Addition is 0 Because if we execute the new Add() ->setValue() function Then we have new object in memory and if we use the new Add()->showAdd() function then we have one more object in memory in show we use the x and y from second object and we not initialize the value for second object in program so we get Addition is 0

If we want to solve this problem we have the two ways

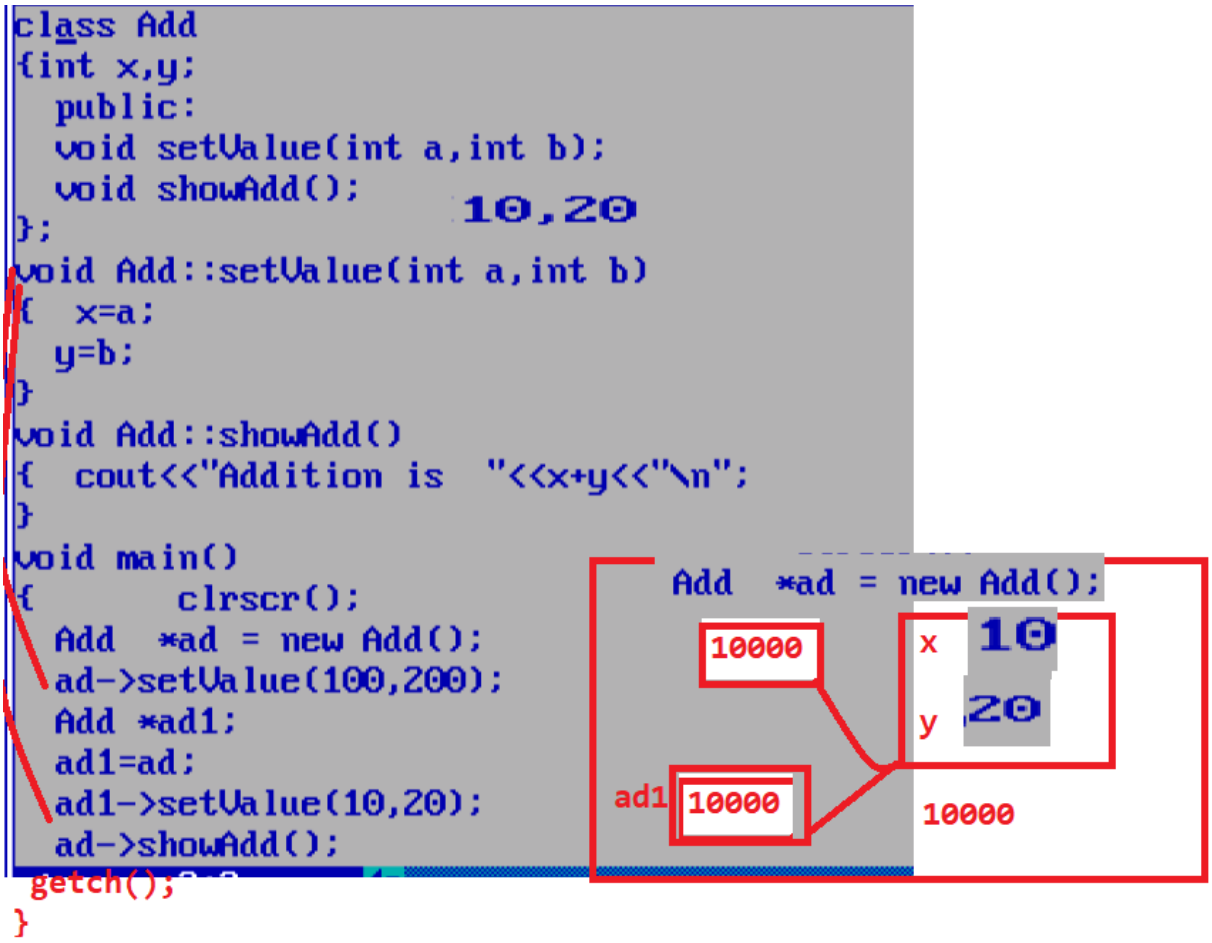
1) we can declare the variable static

2) we can use the reference with object

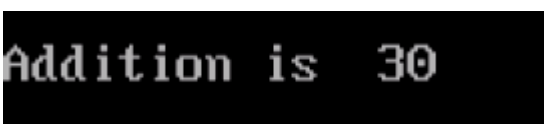
So given diagram shown what happen when we use the reference with object



If we think about above code we have the Add *ad =new Add() means we create the reference of Add class name as *ad and *ad contain the address of object and we use the statement ad->setValue() means we use the object whose address stored ad means as per our example we use the object whose adderss is 10000 and when we use the ad->showAdd() means we use the same object repetitively means here we not create the new object we use the same object two times whose address is 10000



Output:



In think about the above code we have the two different references i.e ad and ad1 here ad and ad1 points to same memory so ad1 override the values on address where ad points previous so when we call the ad->showAdd() we get Addition is 30

What is the diff between object creation without using pointer and with pointer?

Example

---

class Employee {

};

First way of object creation: Employee emp;

Second way of object creation:

Employee *emp = new Employee ();

When we use the first approach with object then we want to work with call by value and we use the second approach when we want to work with call by reference.

Note: for above concept refer video in app Lecture Number 30