Array of objects

```
#include<iostream>
using namespace std;
class Employee
{
int id;
char name[25];
int age;
long salary;
public:
void getdata()
{
cout<<"\n\t enter employee id";cin>>id;
cout<<"\n\t enter employee namw";cin>>name;
cout<<"\n\t enter employee age";cin>>age;
cout<<"\n\t enter employee salary";cin>>salary;
}
void putdata()
{
cout<<"\n"<<id<<"\t"<<name<<"\t"<<age<<"\t"<<salary;
}
};
int main()
{
int i;
Employee E[3];
for( i=0;i<3;i++)
{
  cout<<"\n enter details of"<<i+1<<"\t"<<"Employee";
  E[i].getdata();
  }
  cout<<"\n details of employees";
  for(i=0;i<3;i++)
  E[i].putdata();
  }
```

```
   ^
os@os-HP-Compaq-dc7900-Small-Form-Factor:~$ g++ arr.cpp
os@os-HP-Compaq-dc7900-Small-Form-Factor:~$ ./a.out

 enter details of1      Employee
         enter employee id107

         enter employee namwshreya

         enter employee age23

         enter employee salary45000

 enter details of2      Employee
         enter employee id108

         enter employee namw nita

         enter employee age 21

         enter employee salary78000

 enter details of3      Employee
         enter employee id 109

         enter employee namwpriya

         enter employee age25

         enter employee salary54000

 details of employees
107     shreya  23      45000
108     nita    21      78000
109     priya   25      54000os@os-HP-Compaq-dc7900-Small-Form-Factor:~$
```

THIS POINTER

```
#include<iostream>
#include<string.h>
using namespace std;
class student
{
 int Roll;
 char Name[25];
 float Marks;

 public:
 student(int R,float Mks,char Nm[])
 {
  Roll=R;
  strcpy(Name,Nm);
  Marks=Mks;
 }
 student(char Name[],float Marks,int Roll)
 {
  Roll=Roll;
  strcpy(Name,Name);
  Marks=Marks;
 }
  student(int Roll,char Name[],float Marks)
 {
 this-> Roll=Roll;
  strcpy(this->Name,Name);
 this-> Marks=Marks;
 }
 void display()
 {
 cout<<"\n\tRoll: "<<Roll;
 cout<<"\n\tName: "<<Name;
 cout<<"\n\tMarks: "<<Marks;
  }
 };
 int main()
 {
 student S1(1,89.63,"Sumit");
 student S2("Kumar",78.53,2);
 student S3(3,"Gurav",68.94);

 cout<<"\n\tDetails of student 1: ";
  S1.display();
  cout<<"\n\tDetails of student 2: ";
  S2.display();
  cout<<"\n\tDetails of student 3: ";
  S3.display();
```

```
return 0;
}
```

```
os@os-HP-Compaq-dc7900-Small-Form-Factor:~$ ./a.out

        Details of student 1:
        Roll: 1
        Name: Sumit
        Marks: 89.63
        Details of student 2:
        Roll: 0
        Name:
        Marks: 2.8026e-45
        Details of student 3:
        Roll: 3
        Name: Gurav
        Marks: 68.94os@os-HP-Compaq-dc7900-Small-Form-Factor:~$
```

Dynamic allocation operators

```
#include<iostream>
using namespace std;
int main()
{
double* val=NULL;
 val=new double;
 *val=38184.26;
 cout<<"Value is: "<<*val<<endl;
 delete val;
 return 0;
 }
```

```
os@os-HP-Compaq-dc79  Clone Formatting (double click for mu
os@os-HP-Compaq-dc7900-Small-Form-Factor:~$ g++ dy.cpp
os@os-HP-Compaq-dc7900-Small-Form-Factor:~$ ./a.out
Value is: 38184.3
os@os-HP-Compaq-dc7900-Small-Form-Factor:~$
```