Function overloading

```
#include<iostream>
using namespace std;
class printdata
{
  public:
  void print(int i)
  {
    cout<<"print int:"<<i<<endl;
   }

   void print(double f)
   {
     cout<<"print float:"<<f<<endl;
     }
 };
 int main()
 {
 printdata o;
 o.print(5);
 o.print(27.27);
 return 0;
 }
```

```
os@os-HP-Compaq-dc7900-Small-Form-Factor:~$ g++ fun.cpp
os@os-HP-Compaq-dc7900-Small-Form-Factor:~$ ./a.out
print int:5
print float:27.27
os@os-HP-Compaq-dc7900-Small-Form-Factor:~$
```

Unary operator overloading

```
#include<iostream>
using namespace std;
class Test
{
private:
  int num;
  public:
     Test()
     {
      num=8;
      }
      void operator ++()
      {
       num=--num;
       }
       void print()
       {
        cout<<"the count is : "<<num;
```

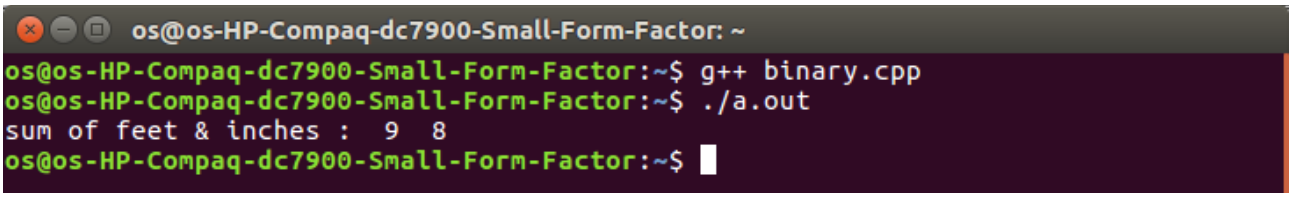
```
    }
};
int main()
{
Test tt;
++tt;
tt.print();
return 0;
}
```

```
os@os-HP-Compaq-dc7900-Small-Form-Factor:~$ g++ unary.cpp
os@os-HP-Compaq-dc7900-Small-Form-Factor:~$ ./a.out
the count is : 7os@os-HP-Compaq-dc7900-Small-Form-Factor:~$
```

Binary operator overloading

```
#include<iostream>
using namespace std;
class Height
{
 public:
 int feet,inch;
 Height()
 {
  feet=0;
  inch=0;
  }
  Height(int f,int i)
  {
   feet=f;
   inch=i;
   }
   Height operator+(Height& d2)
   {
    Height h3;
    h3.feet=feet+d2.feet;
    h3.inch=inch+d2.inch;
    return h3;
    }
    };
    int main()
    {
    Height h1(3,7);
    Height h2(6,1);
    Height h3;
    h3=h1+h2;
    cout<<"sum of feet & inches :  "<<h3.feet<<"  "<<h3.inch<<endl;
    return 0;
```

```
}
```


os@os-HP-Compaq-dc7900-Small-Form-Factor: ~
os@os-HP-Compaq-dc7900-Small-Form-Factor:~$ g++ binary.cpp
os@os-HP-Compaq-dc7900-Small-Form-Factor:~$ ./a.out
sum of feet & inches : 9 8
os@os-HP-Compaq-dc7900-Small-Form-Factor:~$