

Problem-1. Suppose you have a super class named A, which has two variables base, height for two Triangle and you have sub class B. You have to calculate the area of two triangle and compare which triangle area is big use inheritance method.

Code:

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  class A{
4  public:
5      double b1, b2, h1, h2, area1, area2;
6  public:
7      void input(){
8          cin>>b1>>h1>>b2>>h2;
9          area1 = 0.5*b1*h1;
10         area2 = 0.5*b2*h2;
11     }
12 };
13 class B : public A{
14 public:
15     void comp(){
16         if (area1>area2)
17             cout<<"Triangle 1 area is bigger"<<endl;
18         else
19             cout<<"Triangle 2 area is bigger"<<endl;
20     }
21 };
22 int main(){
23     B ob;
24     ob.input();
25     ob.comp();
26     return 0;
27 }
```

Compiled successfully!

Output:

```
5.5 6 10 1.5
Triangle 1 area is bigger
Press any key to continue . . .
```

Discussion:

1. First create a class called **A**(line-3) then declare 6 **public** variables under the class, for two triangle base, height & area .
2. Then read the input these variables value using a public member function **Input** then calculate area for two triangle. Then create a sub class B which inherit public class A, under class B make a comp() function to compare which triangle area is bigger.
4. Create a class object **ob** of class B, B class inherit A class so we can use B class object to access all data of A class. That why ob.input() can take value from user & ob.comp() is compare triangle area & print which is bigger.

Problem-2. Write a C++ program that models a bank account system with classes 'Account', 'Withdraw', and 'Eligible'. Explain how the 'Eligible' class determines if a withdrawal is eligible and what it displays, use inheritance to write the program.

Code:

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  class Account{
4  public:
5      string name;
6      int id,balance;
7      void input1(){
8          cin>>name;
9          cin>>id;
10         cin>>balance;
11     }
12 };
13 class Withdraw{
14 public:
15     int amount;
16     void input2(){
17         cin>>amount;
18     }
19 };
20 class Eligible: public Account,public Withdraw{
21 public:
22     void comp(){
23         if(balance>=amount){
24             cout<<"You can withdraw"<<endl;
25             cout<<"Your balance = "<<balance-amount<<endl;
26         }
27         else{
28             cout<<"You cannot withdraw this amount"<<endl;
29         }
30     }
31 };
32 int main(){
33     while(true){
34         Eligible ob;
35         ob.input1();
36         ob.input2();
37         ob.comp();
38     }
39     return 0;
40 }
```

Output:

```
Nahid——Name
143——ID
1200——Balance
200——Withdraw amount
You can withdraw
Your balance = 1000
```

Discussion:

1. In this code Account & Withdraw is base/super class and Eligible is sub class which inherits Account & Withdraw both class. Account class public function input1() will take value(name, id, main balance) from user and Withdraw class input2() take withdraw amount from user.
2. Then Eligible class inherits Account & Withdraw class (line-20) so, We can use Account & Withdraw class data from Eligible sub class. And here comp() function compare if main balance is greater then withdraw amount then you can withdraw & your balance is balance-amount, else you cannot withdraw.
4. Create a object **ob** of class Eligible, We can access Account & Withdraw data from this object. That why ob.input1() & input2() can take value from user & ob.comp() is compare with balance & amount then print result.

Discussion:

1. Here class A, B & C have a integer array which is ar[10] and B class inherit A class and C class inherit B class so we can access A,B,C all class data from C class object.
2. A class have a int ar[10],sum1 variable & check1() function. Check1() function will read 10 value for ar[10] array also add and store all index value sum1 variable, same as class B & C. Then C class inherits A&B both class so we can access sum1,sum2 from class C. Then compare $\text{sum} - (\text{sum1} * \text{sum2})$ value and store it int diff variable and print this value.
3. From the main function,create C class object ob then call check1 check2, check3 function by ob.check1(), ob.check2(), ob.check3() because C class object ob can access A,B &C both class data.

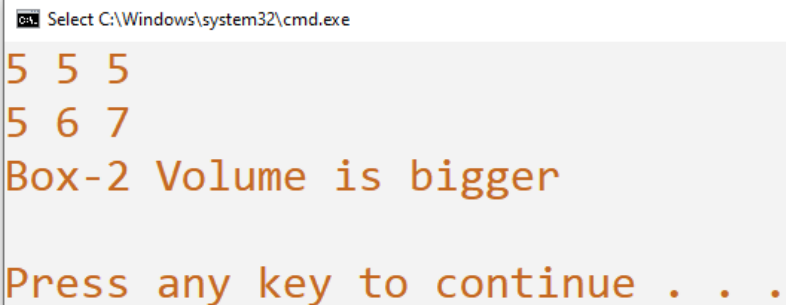
Problem-4. Suppose you have two class named Box1 and Box2, which has three variables height,width and length. You have to calculate the the area of these Box and compare them in a non-member operator function using object or operator overloading function.

Code:

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  class Box2;
4  class Box1{
5  public:
6  int h,w,l,vol1;
7  void input1(){
8  cin>>h >>w >>l;
9  vol1=h*w*l;
10 }
11 friend int operator >(Box1,Box2);
12 };
13 class Box2{
14 public:
15 int h,w,l,vol2;
16 void input2(){
17 cin>>h >>w >>l;
18 vol2=h*w*l;
19 }
20 friend int operator >(Box1, Box2);
21 };

22 int operator>(Box1 o1, Box2 o2){
23 if(o1.vol1>o2.vol2){
24 return 1;
25 }
26 else
27 return 0;
28 }
29 int main(){
30 Box1 ob1;
31 Box2 ob2;
32 ob1.input1();
33 ob2.input2();
34 if(ob1>ob2){
35 cout<<"Box-1 Volume is bigger"<<endl;
36 }
37 else {
38 cout<<"Box-2 Volume is bigger"<<endl;
39 }
40 return 0;
41 }
```

Output:



```
5 5 5
5 6 7
Box-2 Volume is bigger
Press any key to continue . . .
```

Discussion:

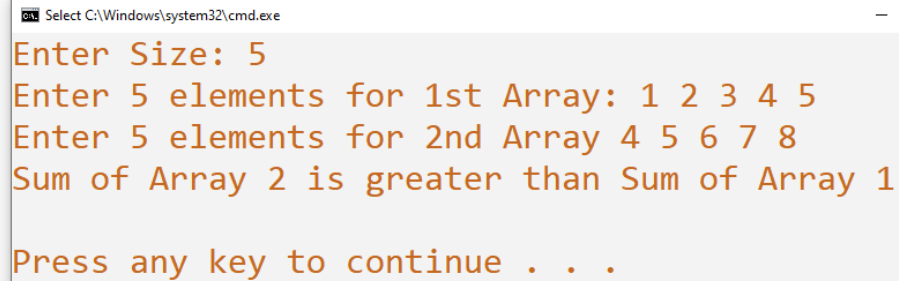
1. Here two classes, **Box1** and **Box2**, are defined. Each class has three integer attributes **h**, **w**, and **l**, representing the height, width, and length of the box, and a **vol1** or **vol2** member variable to store the calculated volume. **input1()** and **input2()** function, which takes user input and calculates the volume.
2. Then Friend Function for Operator Overloading, the **operator >** function is declared as a friend function inside both classes. This function compares two objects of **Box1** and **Box2** based on their volume (**vol1** and **vol2**). It returns **1** if the first box's volume is greater, and **0** otherwise.
3. Then Two objects, **ob1** and **ob2**, of **Box1** and **Box2** are created. And call 2 class input function **ob1.input1()**, **ob2.input2()**; then the overloaded **>** operator is used to compare the volumes of the two boxes, and a message is printed to indicate which box has a larger volume.
4. Finally print "Box-1 Volume is bigger" or "Box-2 Volume is bigger" which box has a larger volume.

Problem-5. Write a Program to compare the sum of the Array of object using object array and object overloading. Show the results.

Code:

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  class MyArray
4  {
5  public:
6      int *arr;
7      int size;
8      MyArray(int n)
9      {
10         size = n;
11         arr = new int[size];
12     }
13     MyArray operator+(MyArray &other)
14     {
15         MyArray result(size);
16         for (int i = 0; i < size; i++)
17         {
18             result.arr[i] = arr[i] + other.arr[i];
19         }
20         return result;
21     }
22     bool operator>(MyArray &other)
23     {
24         int sum1 = 0, sum2 = 0;
25         for (int i = 0; i < size; i++)
26         {
27             sum1 += arr[i];
28             sum2 += other.arr[i];
29         }
30         return sum1 > sum2;
31     }
32     bool operator<(MyArray &other)
33     {
34         int sum1 = 0, sum2 = 0;
35         for (int i = 0; i < size; i++)
36         {
37             sum1 += arr[i];
38             sum2 += other.arr[i];
39         }
40         return sum1 < sum2;
41     }
42     void display()
43     {
44         for (int i = 0; i < size; i++)
45         {
46             cout << arr[i] << " ";
47         }
48         cout << endl;
49     }
50 };
51 int main()
52 {
53     int size;
54     cout << "Enter Size: ";
55     cin >> size;
56     MyArray arr1(size);
57     MyArray arr2(size);
58     cout << "Enter " << size << " elements for 1st Array: ";
59     for (int i = 0; i < size; i++)
60     {
61         cin >> arr1.arr[i];
62     }
63     cout << "Enter " << size << " elements for 2nd Array ";
64     for (int i = 0; i < size; i++)
65     {
66         cin >> arr2.arr[i];
67     }
68     if (arr1 > arr2)
69     {
70         cout << "Sum of Array 1 is greater than Sum of Array 2" << endl;
71     }
72     else if (arr1 < arr2)
73     {
74         cout << "Sum of Array 2 is greater than Sum of Array 1" << endl;
75     }
76     else
77     {
78         cout << "Sum of Array 1 is equal to Sum of Array 2" << endl;
79     }
80     return 0;
81 }
```


Output:



```
Select C:\Windows\system32\cmd.exe
Enter Size: 5
Enter 5 elements for 1st Array: 1 2 3 4 5
Enter 5 elements for 2nd Array 4 5 6 7 8
Sum of Array 2 is greater than Sum of Array 1
Press any key to continue . . .
```

Discussion:

1. In this problem we need to compare 2 array sum, which is bigger use operator overloading concept using c++.
2. The **operator+** overloading method allows you to add two **MyArray** objects element-wise. It creates a new **MyArray** object to store the result and returns it. The **operator>** and **operator<** overloading methods compare the sums of the elements in two **MyArray** objects. They determine whether one array's sum is greater or less than the other, respectively.
3. Then It creates two **MyArray** objects, **arr1** and **arr2**, of the same size. Then compares the sums of the elements in these arrays and prints the result of the comparison.