```cpp
#include <iostream>
using namespace std;

class SumNatural {
    int n, sum;
public:
    SumNatural() {
        cout << "Enter a number:
";
        cin >> n;
        sum = (n * (n + 1)) / 2;
    }
    void display() {
        cout << "Sum of first "
<< n << " natural numbers is: "
<< sum << endl;
    }
};

int main() {
    SumNatural obj;
    obj.display();
    return 0;
}
```

```cpp
#include <iostream>
using namespace std;

class SwapNumbers {
    int a, b;
public:
    SwapNumbers(int x, int y) :
a(x), b(y) {
        cout << "Before swapping:
a = " << a << ", b = " << b <<
endl;
        int temp = a;
        a = b;
        b = temp;
    }
    void display() {
        cout << "After swapping:
a = " << a << ", b = " << b <<
endl;
    }
};

int main() {
    SwapNumbers obj(5, 10);
    obj.display();
    return 0;
}
```

```cpp
#include <iostream>
using namespace std;

class SumArray {
    int *arr, n, sum;
public:
    SumArray(int a[], int size) :
n(size), sum(0) {
        arr = new int[n];
        for (int i = 0; i < n; i+
+) {
            arr[i] = a[i];
            if (arr[i] > 0) sum
+= arr[i];
        }
    }
    void display() {
        cout << "Sum of all
positive numbers in the array is:
" << sum << endl;
    }
    ~SumArray() { delete[] arr; }
};

int main() {
    int arr[] = {1, -2, 3, 4,
-5};
    SumArray obj(arr, 5);
    obj.display();
    return 0;
}
```

```cpp
#include <iostream>
using namespace std;

class SumValues {
public:
    SumValues(int a, int b) {
        cout << "Sum of integers:
" << a + b << endl;
    }
    SumValues(float a, float b) {
        cout << "Sum of floats: "
<< a + b << endl;
    }
    SumValues(char a, char b) {
        cout << "Sum of chars
(ASCII): " << int(a) + int(b) <<
endl;
    }
};

int main() {
    SumValues obj1(5, 10);
    SumValues obj2(5.5f, 10.5f);
    SumValues obj3('a', 'b');
    return 0;
}
```

```cpp
#include <iostream>
using namespace std;

class Base {
protected:
    int num1;
public:
    Base(int x) : num1(x) {}
};

class Derived : public Base {
    int num2;
public:
    Derived(int x, int y) :
Base(x), num2(y) {}
    void displayProduct() {
        cout << "Product of
numbers: " << num1 * num2 <<
endl;
    }
};

int main() {
    Derived obj(5, 10);
    obj.displayProduct();
    return 0;
}
```

```cpp
#include <iostream>
using namespace std;

class Base {
protected:
    int num1;
public:
    Base(int x) : num1(x) {}
};

class Derived1 : public Base {
protected:
    int num2;
public:
    Derived1(int x, int y) :
Base(x), num2(y) {}
};

class Derived2 : public Derived1
{
public:
    Derived2(int x, int y) :
Derived1(x, y) {}
    void displaySum() {
        cout << "Sum of numbers:
" << num1 + num2 << endl;
    }
};

int main() {
    Derived2 obj(5, 10);
    obj.displaySum();
    return 0;
}
```

```cpp
#include <iostream>
using namespace std;

class Base1 {
protected:
    int num1;
public:
    Base1(int x) : num1(x) {}
};

class Base2 {
protected:
    int num2;
public:
    Base2(int y) : num2(y) {}
};

class Derived : public Base1,
public Base2 {
public:
    Derived(int x, int y) :
Base1(x), Base2(y) {}
    void displaySum() {
        cout << "Sum of numbers:
" << num1 + num2 << endl;
    }
};

int main() {
    Derived obj(5, 10);
    obj.displaySum();
    return 0;
}
```

```cpp
#include <iostream>
using namespace std;

class Base {
protected:
    int num1, num2;
public:
    Base(int x, int y) : num1(x),
num2(y) {}
};

class Derived1 : public Base {
public:
    Derived1(int x, int y) :
Base(x, y) {}
    void displayFirst() {
        cout << "First number: "
<< num1 << endl;
    }
};

class Derived2 : public Base {
public:
    Derived2(int x, int y) :
Base(x, y) {}
    void displaySecond() {
        cout << "Second number: "
<< num2 << endl;
    }
};
```

```
int main() {
    Derived1 obj1(5, 10);
    Derived2 obj2(5, 10);
    obj1.displayFirst();
    obj2.displaySecond();
    return 0;
}
```

```cpp
#include <iostream>
using namespace std;

class A {
public:
    void display() {
        cout << "Class A" <<
endl;
    }
};

class B : public A {
public:
    void displayB() {
        cout << "Class B" <<
endl;
    }
};

class C : public A {
public:
    void displayC() {
        cout << "Class C" <<
endl;
    }
};

class D : public B, public C {
public:
    void displayD() {
        cout << "Class D" <<
```

```
endl;
    }
};

int main() {
    D objD;
    objD.display();
    objD.displayB();
    objD.displayC();
    objD.displayD();

    E objE;
    objE.display();
    objE.displayB();
    objE.displayE();

    return 0;
}
```