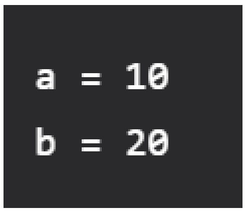
1. Write a C++ program to an abstract class which shown the concept of abstraction.

```
#include <iostream>

using namespace std;


class implementAbstraction

{

        private:

                int a, b;


        public:


                // method to set values of

                // private members

                void set(int x, int y)

                {

                        a = x;

                        b = y;

                }


                void display()

                {

                        cout<<"a = " <<a << endl;

                        cout<<"b = " << b << endl;

                }

};


int main()

{

        implementAbstraction obj;

        obj.set(10, 20);

        obj.display();
        return 0;
}
```

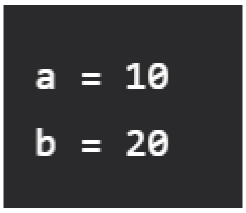


```
a = 10
b = 20
```

2. Write a C++ program where you implement a class with public and private members is an example of data abstraction.

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

class implementAbstraction
{
        private:
                int a, b;

        public:

                // method to set values of
                // private members
                void set(int x, int y)
                {
                        a = x;
                        b = y;
                }

                void display()
                {
                        cout<<"a = " <<a << endl;
                        cout<<"b = " << b << endl;
```

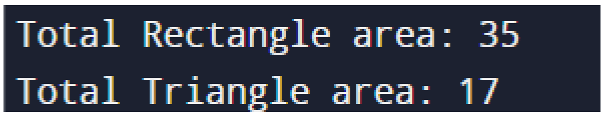
```
                    }
};

int main()
{
        implementAbstraction obj;
        obj.set(10, 20);
        obj.display();
        return 0;
}
```



```
a = 10
b = 20
```

3. Write a C++ program to calculate the area of a rectangle and triangle using the abstract class.

```
#include <iostream>

using namespace std;

// Base class
class Shape {
  public:
    // pure virtual function providing interface framework.
    virtual int getArea() = 0;
    void setWidth(int w) {
      width = w;
    }

    void setHeight(int h) {
      height = h;
    }
  protected:
    int width;
    int height;
};

// Derived classes
class Rectangle: public Shape {
  public:
    int getArea() {
      return (width * height);
    }
};

class Triangle: public Shape {
  public:
    int getArea() {
      return (width * height)/2;
    }
};

int main(void) {
  Rectangle Rect;
  Triangle  Tri;

  Rect.setWidth(5);
  Rect.setHeight(7);

  // Print the area of the object.
  cout << "Total Rectangle area: " << Rect.getArea() << endl;
  Tri.setWidth(5);
  Tri.setHeight(7);

  // Print the area of the object.
  cout << "Total Triangle area: " << Tri.getArea() << endl;

  return 0;
}
```

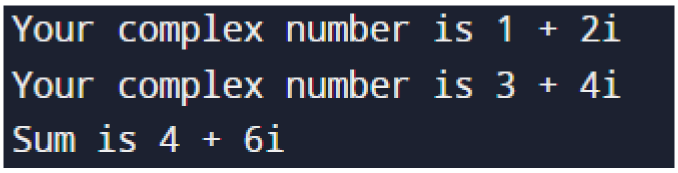


```
Total Rectangle area: 35
Total Triangle area: 17
```

5. Write a complex class which hide the complexity of adding two number and Add two Complex Numbers by Passing to a member function.

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

class complex{
   int a;
   int b;

   public:
      void setData(int v1, int v2){
         a = v1;
         b = v2;
      }

      void setDataBySum(complex o1, complex o2){
         a = o1.a + o2.a;
         b = o1.b + o2.b;
      }
```

```
        void printNumber(){
            cout<<"Your complex number is "<<a<<" + "<<b<<"i"<<endl;
        }
        void sum(){
            cout<<"Sum is "<<a<<" + "<<b<<"i"<<endl;
        }
};

int main(){
    complex c1, c2, c3;
    c1.setData(1, 2);
    c1.printNumber();

    c2.setData(3, 4);
    c2.printNumber();

    c3.setDataBySum(c1, c2);
    c3.sum();
    return 0;
}
```



```
Your complex number is 1 + 2i
Your complex number is 3 + 4i
Sum is 4 + 6i
```