//programme 1

/* "Declare a class called bird having private data members name and weight. Define following functions :

- default constructor for reading data members from keyboard

- overloaded constructor with two arguments to be used for initialization of data members.

- display function to display data members.

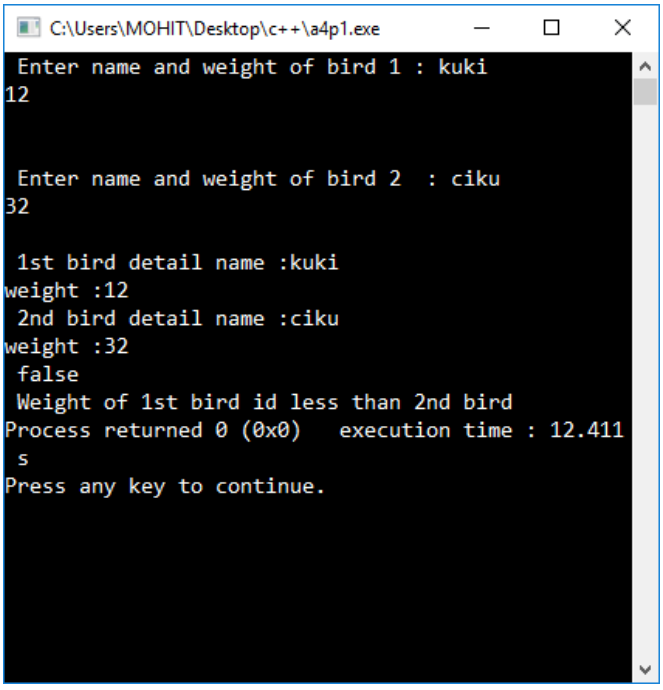
- overloaded member operator >= to compare weight of two bird objects, returning false if weight of first bird object is less than that of the second & true otherwise. Define main to illustrate use of above functions."

*/

```
#include<stdio.h>
#include<iostream>
#include<string.h>
using namespace std;
class bird{
private:
   char name[50];
   float weight;
public:
   bird(){}
   bird(char a[],float w){
   strcpy(name,a);
   weight=w;
```

```
    }
    void display();
    int operator >=(bird);
};
void bird:: display(){
    cout << "name :" << name << endl << "weight :" << weight ;
}
int bird :: operator >=(bird b)
{
        if(weight< b.weight){
                cout  << "\n false ";
            return 0;}
        else{
                cout << "\n true ";
            return 1;}
}
int main()
{
    bird b1,b2;
    int b1_weight,b2_weight;
    char nameb1[50],nameb2[50];
        cout << "\n Enter name and weight of bird 1 : ";
            cin >> nameb1 >> b1_weight;
                cout << endl;
        cout << "\n Enter name and weight of bird 2  : ";
            cin >> nameb2 >> b2_weight;
    b1=bird(nameb1,b1_weight);
```

```
    b2=bird(nameb2,b2_weight);
      cout << "\n 1st bird detail ";
         b1.display();
      cout << "\n 2nd bird detail ";
         b2.display();
         if(b1>=b2)
         {
            cout << "\n Weight of 2nd bird id less than 1st bird ";
      }
         else{
               cout << "\n Weight of 1st bird id less than 2nd bird ";
    }
         return 0;
}
```



// programme 2

/*"Define a class complex with real and imaginary as two data member, add

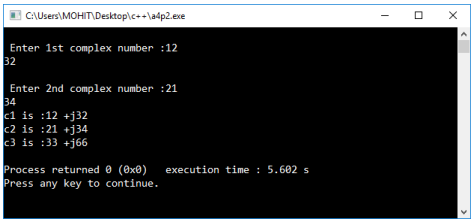
necessary constructors and member function to initialize and display data of

class. Class should overload the + operator to add two complex objects and

return the results. Invoke the statements like C3=C1+C2 in main ()." */

```
#include<iostream>
using namespace std;
class complex{
   float x,y;
public:
   complex(){}
   complex(float real,float imag){
      x=real;
      y=imag;
   }
   complex operator +(complex);
   void display();
};
complex complex :: operator +(complex c){
      complex temp;
```

```
        temp.x=x+c.x;
        temp.y=y+c.y;
        return temp;
}
void complex:: display()
{
    cout << x << " +j" << y;
}
int main()
{
    int real,imag;
    complex c3;
        cout << "\n Enter 1st complex number :";
            cin >> real >> imag;
                complex c1(real,imag);
        cout << "\n Enter 2nd complex number :";
            cin >> real >> imag;
                complex c2(real,imag);
                c3=c1+c2;
                    cout << "c1 is :" ;  c1.display(); cout << endl;
                    cout << "c2 is :" ; c2.display(); cout << endl;
                    cout << "c3 is :" ; c3.display(); cout << endl;
                    return 0; }
```



// programme 3

/* Declare a class called book having members like book_title, publisher and author_name. Overload extractor and inserter operators (>> and <<) for class book.

*/

```
#include<iostream>
using namespace std;
class book{
      string book_title;
      string publisher;
      string author_name;
   public:
         friend istream& operator >>(istream&,book&);
         friend ostream& operator <<(ostream&,book&);
};
istream & operator >> (istream &in, book &b1){

      in >> b1.book_title >> b1.publisher >> b1.author_name;
}
ostream & operator << (ostream &out, book &b1){
      out << "\n book title" << b1.book_title << "\n author name " << b1.author_name << "\n publisher " << b1.publisher ;
}
int main()
{
book b;
```

```
        cout << "\nEnter Book Title, Publisher name and author name: ";
        cin >> b;
        cout << "\nBook Details are as shown below:";
        cout << b;
        return 0;
}
```