```
#include<iostream>
using namespace std;
class Complex
{
   public:
   float real,img;
   Complex()
   {
      real=0;
      img=0;
   }
   Complex(int r, int i)
   {
      real=r;
      img=i;
   }
   Complex operator+(Complex obj)
   {
      Complex temp;
      temp.real=real+obj.real;
      temp.img=img+obj.img;
      return temp;
   }
   Complex operator*(Complex obj)
   {
      Complex temp;
      temp.real = (real*obj.img - img*obj.img);
      temp.img  = (img*obj.real + real*obj.img);
      return temp;
   }
};
int main()
{
   Complex c1(20,40);
   Complex c2(10,10);
   Complex c3;
   c3=c1+c2;
   cout<<"the sum of numbers is : ";
   cout<<c3.real<<'+'<<c3.img<<'i'<<endl;
   c3=c1*c2;
```

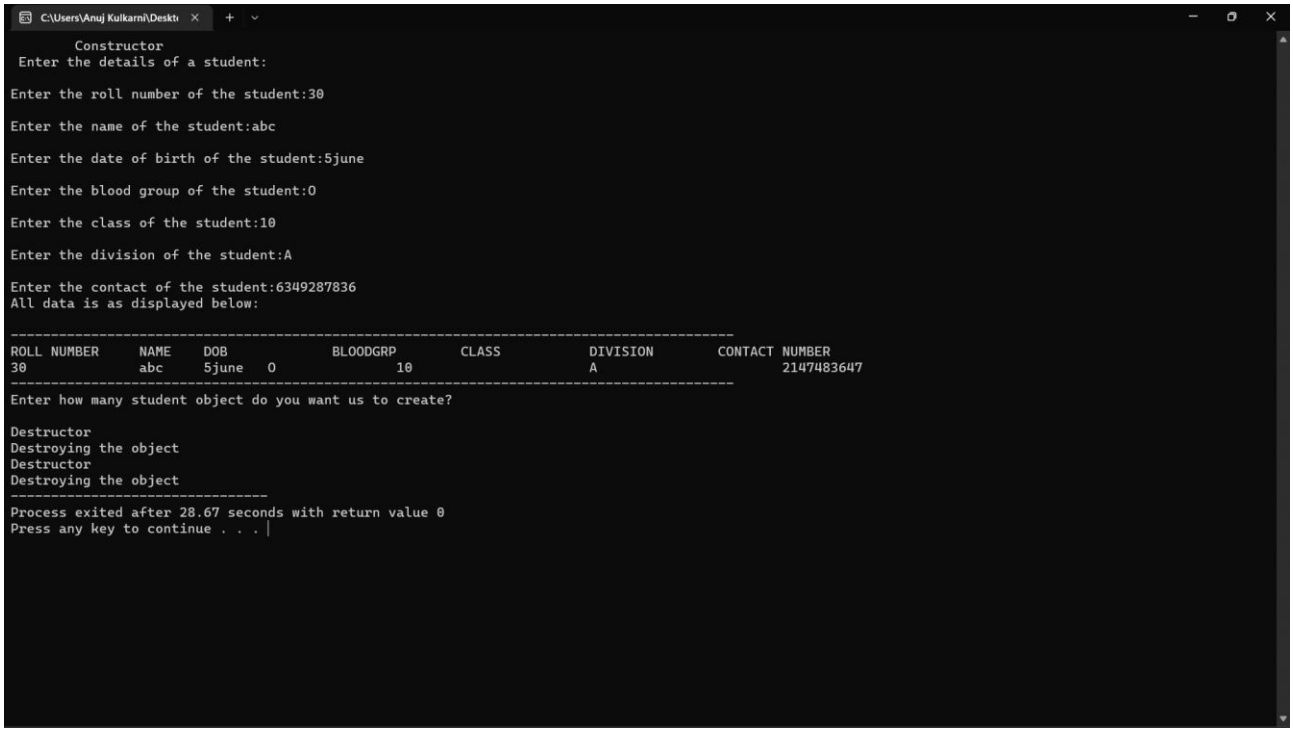
```
    cout<<"multiplication of numbers is : ";
    cout<<c3.real<<'+'<<c3.img<<'i'<<endl;
    return 0;
}
```

```
#include <iostream>
#include<string.h>
using namespace std;
class student
{
public:
int roll_no;
char clas[10];
int sr_no;
long int tele_no;
char name[20];
char div;
char blood_grp[20];
char DOB[10];
static int count;
void getdata();
friend void display(student & obj);
student() //Constructor
{
roll_no=0;
cout<<"\tConstructor";
roll_no=count;
count++;
}
~student() //Destructor
{
cout<<"\nDestructor";
cout<<"\nDestroying the object";
count--;
}
student(int roll_no)
{
this->roll_no=roll_no;
}
student (student & obj)
{
roll_no=obj.roll_no;
strcpy(name,obj.name);
strcpy(DOB,obj.DOB);
strcpy(clas,obj.clas);
strcpy(blood_grp,obj.blood_grp);
div=obj.div;
tele_no=obj.tele_no;
sr_no=count;
count++;
}
};
int student :: count=0;
```

```
void student:: getdata()
{
cout<<"\n"<<"Enter the roll number of the student:";
cin>>roll_no;
cout<<"\n"<<"Enter the name of the student:";
cin>>name;
cout<<"\n"<<"Enter the date of birth of the student:";
cin>>DOB;
cout<<"\n"<<"Enter the blood group of the student:";
cin>>blood_grp;
cout<<"\n"<<"Enter the class of the student:";
cin>>clas;
cout<<"\n"<<"Enter the division of the student:";
cin>>div;
cout<<"\n"<<"Enter the contact of the student:";
cin>>tele_no;
}
void display(student & obj)
{
cout<<"\n"<<obj.roll_no;
cout<<"\t\t"<<obj.name;
cout<<"\t"<<obj.DOB;
cout<<"\t"<<obj.blood_grp;
cout<<"\t\t"<<obj.clas;
cout<<"\t\t\t"<<obj.div;
cout<<"\t\t\t"<<obj.tele_no;
}
int main()
{
student s1;
student s2(s1);
cout<<"\n Enter the details of a student:"<<"\n";
s1.getdata();
cout<<"All data is as displayed below:"<<"\n";
cout<<"\n--------------------------------------------------------------------------------------------";
cout<<"\nROLL
NUMBER\tNAME\tDOB\t\tBLOODGRP\tCLASS\t\tDIVISION\tCONTACT
NUMBER";
display(s1);
cout<<"\n--------------------------------------------------------------------------------------------";
int i,n;
student *s[50];
cout<<"\nEnter how many student object do you want us to create?"<<"\n";
cin>>n;
for(i=0;i<n;i++)
{
s[i]= new student();
}
```

```
for(i=0;i<n;i++)
{
s[i]->getdata();
}
for(i=0;i<n;i++)
{
display(*s[i]);
}
for(i=0;i<n;i++)
{
delete (s[i]);
}
return 0;
}
```

OUTPUT :-



```
#include<iostream>
#include<string.h>
#include<exception>
using namespace std;
class publication
{
        public:
                string title;
                float price;
                void getdata()
                {
                        cout<<"Enter the name of book : ";
                        cin>>title;
                        cout<<"Enter the cost of book : ";
                        cin>>price;
                }
                void showdata()
                {
                        cout<<"Name of book is : "<<title<<endl;
                        cout<<"Cost of book is : "<<price<<endl;
                }
};
class book:public publication
{
   public:
                int count;
                void getdata()
                {
                        cout<<"Enter the number of pages in the book : ";
                        cin>>count;
                }
                void showdata()
                {
                        try
                        {
                                if(count<=0)
                                {
                                        throw count;
                                }
                                else
```

```
                {
                        cout<<"Pages of book are : "<<count<<endl;
                }
        }
        catch(int i)
        {
                cout<<"Enter page number greater than 0 \n";
                count=0;
        }
    }
};
class tape:public publication
{
   public:
        float min;
        void getdata()
        {
                cout<<"Enter the time of the audio book : ";
                cin>>min;
        }
        void showdata()
        {
                try
                {
                        if(min<=0)
                        {
                                throw min;
                        }
                        else
                        {
                                cout<<"Time of audio book is : "<<min<<endl;
                        }
                }
                catch(float i)
                {
                        cout<<"Enter audio tape greater than 0 ";
                        min=0;
                }
        }
};
```

```
int main()
{
        publication p;
        book b;
        tape t;
        p.getdata();
        p.showdata();
        b.getdata();
        b.showdata();
        t.getdata();
        t.showdata();
          return 0;
 }
```

OUTPUT :-

```
#include<iostream>
#include<fstream>
using namespace std;
class test
{
   public:
          void writedata();
          void readdata();
};
void test::writedata()
{
   fstream fp;
   char ch;
   fp.open("it.txt",ios::out);
   cin>>ch;
   while (ch!='.')
   {
      fp.put(ch);
      cin>>ch;
   }
   fp.close();
}
void test::readdata()
{
          fstream fp;
          char ch;
          fp.open("it.txt",ios::in);
   ch=fp.get();
   while(!fp.eof())
   {
      cout<<ch;
      ch=fp.get();
   }
   fp.close();
}
int main()
{
   test ob;
   int ch;
   do
   {
      cout<<"\n1.Write\n2.Read\n3.Exit";
      cout<<"\nEnter your choice= ";
      cin>>ch;
      switch(ch)
      {
         case 1:
```

```
        ob.writedata();
        break;
        case 2:
              ob.readdata();
        break;
        case 3:
        break;
      }
   }
   while(ch<3);
   return 0;
}
```

OUTPUT :-

```
#include<iostream>
using namespace std;
template<class T>
void swapping(T &a , T &b)
{
   int temp;
   temp=a;
   a=b;
   b=temp;
}
void display(int *array , int size)
{
   for(int i = 0; i<size; i++)
   {
     cout << array[i] << " ";
     cout << endl;
   }
}
void selection_sort(int *array , int size)
{
   int i,j,min;
   for (i=0;i<size-1;i++)
   {
      min=i;
      for (j=i+1;j<size;j++)
      {
         if(array[j]<array[min])
         {
            min=j;
         }
         swap(array[i],array[min]);
      }
   }
}
int main()
{
   int n;
   cout<<"number of element in the list : \n";
   cin>>n;
   int arr[n];
   cout<<"enter the elements : \n";
   for(int i=0;i<n;i++)
   {
      cin>>arr[i];
   }
   cout<<"element before sorting :"<<endl;
   display(arr,n);
```

```
    selection_sort(arr,n);
    cout<<"element after sorting :"<<endl;
    display(arr,n);
    return 0;
}
```

OUTPUT :-

```
#include <iostream>
#include <algorithm>
#include <vector>
using namespace std;
class Item
{
   public:
   char name[10];
   int quantity;
   int cost;
   int code;
   bool operator==(const Item& i1)
   {
      if(code==i1.code)
      return 1;
      return 0;
   }
   bool operator<(const Item& i1)
   {
      if(code<i1.code)
      return 1;
      return 0;
   }
};
vector<Item>o1;
void print(Item &i1);
void display();
void insert();
void search();
void dlt();
bool compare(const Item &i1, const Item &i2)
{
   return i1.cost < i2.cost;
}
int main()
{
   int ch;
   do
   {
      cout<<"\n* * * * * Menu * * * * *";
```

```
        cout<<"\n1. Insert\n2. Display\n3. Search\n4. Sort\n5. Delete\n6. Exit";
        cout<<"\nEnter your choice : ";
        cin>>ch;
        switch(ch)
        {
            case 1:
            insert();
            break;

            case 2:
            display();
            break;

            case 3:
            search();
            break;

            case 4:
            sort(o1.begin(),o1.end(),compare);
            cout<<"\n\n Sorted on Cost : ";
            display();
            break;

            case 5:
            dlt();
            break;

            case 6:
            exit(0);
        }
    }
    while(ch!=7);
    return 0;
}
void insert()
{
    Item i1;
    cout<<"Enter Item Name : ";
    cin>>i1.name;
    cout<<"Enter Item Quantity : ";
```

```
    cin>>i1.quantity;
    cout<<"Enter Item Cost : ";
    cin>>i1.cost;
    cout<<"Enter Item Code : ";
    cin>>i1.code;
    o1.push_back(i1);
}
void display()
{
    for_each(o1.begin(),o1.end(),print);
}
void print(Item &i1)
{
    cout<<"\n";
    cout<<"\nItem Name : "<<i1.name;
    cout<<"\nItem Quantity : "<<i1.quantity;
    cout<<"\nItem Cost : "<<i1.cost;
    cout<<"\nItem Code : "<<i1.code;
    cout<<"\n\n";
}
void search()
{
    vector<Item>::iterator p;
    Item i1;
    cout<<"\nEnter Item Code to search : ";
    cin>>i1.code;
    p=find(o1.begin(),o1.end(),i1);
    if(p==o1.end())
    {
        cout<<"\nNot found!!!";
    }
    else
    {
        cout<<"\nFound!!!";
    }
}
void dlt()
{
    vector<Item>::iterator p;
    Item i1;
```

```
    cout<<"\nEnter Item Code to delete : ";
    cin>>i1.code;
    p=find(o1.begin(),o1.end(),i1);
    if(p==o1.end())
    {
        cout<<"\nNot found!!!";
    }
    else
    {
        o1.erase(p);
        cout<<"\nDeleted!!!";
    }
}
```

OUTPUT :-

```
#include<iostream>
#include<map>
#include<string>
using namespace std;
int main()
{
typedef map<string,int> mapType;
mapType populationMap;
populationMap.insert(pair<string, int>("Maharashtra", 7026357));
populationMap.insert(pair<string, int>("Rajasthan", 6578936));
populationMap.insert(pair<string, int>("Karanataka", 6678993));
populationMap.insert(pair<string, int>("Punjab", 5789032));
populationMap.insert(pair<string, int>("West Bengal", 6676291));
mapType::iterator iter;
cout<<"========Population of states in India==========\n";
cout<<"\n Size of populationMap"<<populationMap.size()<<"\n";
string state_name;
cout<<"\n Enter name of the state :";
cin>>state_name;
iter = populationMap.find(state_name);
if( iter!= populationMap.end() )
 cout<<state_name<<" 's population is "
 <<iter->second ;
else
cout<<"Key is not populationMap"<<"\n";
populationMap.clear();
}
```

OUTPUT :-


C:\Users\Anuj Kulkarni\Deskt
========Population of states in India==========
Size of populationMap5
Enter name of the state :Maharashtra
Maharashtra 's population is 7026357

Process exited after 9.292 seconds with return value 0
Press any key to continue . . .