**Assignment No.1**

**Program:**

#include<iostream.h>

#include<conio.h>

#include<stdio.h>

class complex

{

public:

float real,imag;

complex()

{

real=0;

imag=0;

}

complex operator+(complex);

complex operator\*(complex) ;

friend ostream& operator<<(ostream&,complex&);

friend istream& operator>>(istream&,complex&);

};

complex complex::operator+(complex obj)

{

complex temp;

temp.real=real+obj.real;

temp.imag=imag+obj.imag;

return(temp);

}

complex complex::operator\*(complex obj)

{

complex temp;

temp.real=(real\*obj.real)-(imag\*obj.imag);

temp.imag=(real\*obj.imag)+(obj.real\*imag);

return (temp);

}

istream& operator>>(istream& is,complex& obj)

{

is>>obj.real;

is>>obj.imag;

return is;

}

ostream& operator<<(ostream& os,complex& obj)

{

os<<obj.real;

os<<"+"<<obj.imag<<"i";

return os;

}

int main()

{

clrscr();

complex a,b,c,d;

cout<<"\n Default value of complex no.is="<<a;

cout<<"\n Enter first complex no.(real and imaginary part)";

cin>>a;

cout<<"\nfirst no is="<<a;

cout<<"\n Enter second complex no.(real and imaginary part)";

cin>>b;

cout<<"\n second no.is="<<b;

cout<<"\n Arithmatic operations are:";

c=a+b;

cout<<"\n Addition is=" <<c;

d=a\*b;

cout<<"\n Multiplication is"<<d;

getch();

return 0;

}

**Output:**

Enter the 1st number

Enter the real part = 2

Enter the imaginary part=4

Enter the 2nd number

Enter the real part=4

Enter the imaginary part=8

The first number is =2+4i

The second number is= 4+8i

The addition is =6+12i

The multiplication is= -24+32i

**Assignment No.2**

**Program:**

#include<iostream>

#include<string.h>

using namespace std;

class student

{

char name[10];

int rollno;

char class\_name[3];

char address[10];

long int tellno;

long int licno;

char blood\_group[10];

int dob;

public:

student()

{

strcpy(name," ABC");

rollno=6;

strcpy(class\_name,"SE");

strcpy(address,"Malegaon");

tellno=1234567889;

licno=23456779;

strcpy(blood\_group,"A+ve");

dob=2004;

}

student(char n[],int r,char c[],char a[],int t,int l,char b[],int d)

{

strcpy(name,n);

rollno=r;

strcpy(class\_name,c);

strcpy(address,a);

tellno=t;

licno=l;

strcpy(blood\_group,b);

dob=d;

}

student(const student & x)

{

strcpy(name,x.name);

rollno=x.rollno;

strcpy(class\_name,x.class\_name);

strcpy(address,x.address);

tellno=x.tellno;

licno=x.licno;

strcpy(blood\_group,x.blood\_group);

dob=x.dob;

}

void display();

};

void student::display()

{

cout<<"\n name="<<name;

cout<<"\n rollno="<<rollno;

cout<<"\n class name"<<class\_name;

cout<<"\n address="<<address;

cout<<"\n tellno"<<tellno;

cout<<"\n licno"<<licno;

cout<<"\n blood\_group"<<blood\_group;

cout<<"\n dob="<<dob;

};

int main()

{

student dobj;

cout<<"\n default constructor";

dobj.display();

cout<<"\n parametarized constructor"<<"\n enter parameters";

char pname[20],pclass\_name[3],paddress[20],pblood\_group[10];

int prollno,pdob;

long int ptellno,plicno;

cout<<"\n enter name:";

cin>>pname;

cout<<"\n enter class\_name:";

cin>>pclass\_name;

cout<<"\n enter address:";

cin>>paddress;

cout<<"\n blood group:";

cin>>pblood\_group;

cout<<"\n enter roll no:";

cin>>prollno;

cout<<"\n enter dob:";

cin>>pdob;

cout<<"\n enter tellno:";

cin>>ptellno;

cout<<"\n licno:";

cin>>plicno;

student pobj(pname,prollno,pclass\_name,paddress,ptellno,plicno,pblood\_group,pdob);

pobj.display();

cout<<"\n copy constructor";

student newobj=pobj;

newobj.display();

return 0;

}

**Output:**

default constructor

name= ABC

rollno=6

class nameSE

address=Malegaon

tellno1234567889

licno23456779

blood\_groupA+ve

dob=2004

parametarized constructor

enter parameters

enter name:Payal

enter class\_name:SE

enter address:Malegaon

blood group:A+

enter roll no:206

enter dob:2004

enter tellno:266355627

licno:225637487596

name=Payal

rollno=206

class nameSE

address=Malegaon

tellno266355627

licno2147483647

blood\_groupA+

dob=2004

copy constructor

name=Payal

rollno=206

class nameSE

address=Malegaon

tellno266355627

licno2147483647

blood\_groupA+

dob=2004

**Assignment No.3**

**Program:**

#include <iostream>

#include<string>

using namespace std;

class publication

{

protected:

string title;

float price;

public:

publication()

{

title=" ";

price=0.0;

}

publication(string t,float p)

{

title=t;

price=p;

}

};

class book : public publication

{

int pagecount;

public:

book()

{

pagecount=0;

}

//After : base class constructor is called

book(string t,float p,int pc):publication(t,p)

{

pagecount=pc;

}

void display()

{

cout<<"title :"<<title<<endl;

cout<<"Price: "<<price<<endl;

cout<<"Pagecount :"<<pagecount<<endl;

}

};

class CD : public publication

{

float time;

public:

CD()

{

time=0.0;

}

//After : base class constructor is called

CD(string t,float p,float tim):publication(t,p)

{

time=tim;

}

void display()

{

cout<<"title :"<<title<<endl;

cout<<"Price: "<<price<<endl;

cout<<"time in minutes :"<<time<<endl;

}

};

int main()

{

cout<<endl<<"Book data"<<endl;

book b("C++",230,300);

b.display();

cout<<endl<<"CD Data"<<endl;

CD c("programming",100,120.5);

c.display();

return 0;

}

**Output:**

Book data

title :C++

Price: 230

Pagecount :300

CD Data

title :programming

Price: 100

time in minutes :120.5

**Assignment No.4**

**Program:**

#include <iostream>

#include <fstream>

using namespace std;

class file

{

char name[40];

int emp\_id;

float salary;

public:

void accept()

{

cin>>name;

cin>>emp\_id;

cin>>salary;

}

void display()

{

cout<<"\n"<<name<<"\t"<<emp\_id<<"\t"<<salary;

}

};

int main()

{

file obj[5];

fstream f;

int i,n;

f.open("input.txt", ios::out);

cout<<"\nHow many employee information want to store ";

cin>>n;

cout<<"\nEnter information of employee";

for(i=0;i<n;i++)

{

cout<<"\nEnter information of "<<i+1<<" employee::";

obj[i].accept();

f.write((char\*) &obj[i],sizeof(obj[i]));

}

f.close();

f.open("input", ios::in);

cout<<"\nEntered information of employee is:: ";

for(i=0;i<n;i++)

{

f.read((char\*) &obj[i],sizeof(obj[i]));

obj[i].display();

}

f.close();

return 0;

}

**Output:**

How many employee information want to store 3

Enter information of employee

Enter information of 1 employee::rakesh

1201

25000

Enter information of 2 employee::mukesh

1202

26000

Enter information of 3 employee::sunil

01 1203

28000

Entered information of employee is::

rakesh 1201 25000

mukesh 1202 26000

sunil 1203 28000

**Assignment No.5**

**Program:**

#include<iostream>

using namespace std;

int n;

#define size 10

template<class T>

void sel(T A[size])

{

int i,j,min;

T temp;

for(i=0;i<n-1;i++)

{

min=i;

for(j=i+1;j<n;j++)

{

if(A[j]<A[min])

min=j;

}

temp=A[i];

A[i]=A[min];

A[min]=temp;

}

cout<<"\nSorted array:";

for(i=0;i<n;i++)

{

cout<<" "<<A[i];

}

}

int main()

{

int A[size];

float B[size];

int i;

cout<<"\nEnter total no of int elements:";

cin>>n;

cout<<"\nEnter int elements:";

for(i=0;i<n;i++)

{

cin>>A[i];

}

sel(A);

cout<<"\nEnter total no of float elements:";

cin>>n;

cout<<"\nEnter float elements:";

for(i=0;i<n;i++)

{

cin>>B[i];

}

sel(B);

}

**Output:**

Enter total no of int elements:4

Enter int elements:12

15

66

54

Sorted array: 12 15 54 66

Enter total no of float elements:4

Enter float elements:4.2

1.5

6.3

9.5

Sorted array: 1.5 4.2 6.3 9.5

**Assignment No.6**

**Program:**  
#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";

cout<<"\n2.Display";

cout<<"\n3.Search";

cout<<"\n4.Sort";

cout<<"\n5.Delete";

cout<<"\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<<"\nEnter Item Name:";

cin>>i1.name;

cout<<"\nEnter Item Quantity:";

cin>>i1.quantity;

cout<<"\nEnter Item Cost:";

cin>>i1.cost;

cout<<"\nEnter 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;

}

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."<<endl;

cout<<"Item Name : "<<p ->name<<endl;

cout<<"Item Quantity : "<<p ->quantity<<endl;

cout<<"Item Cost : "<<p ->cost<<endl;

cout<<"Item Code: "<<p ->code<<endl;

}

}

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:**

\*\*\* Menu \*\*\*

1.Insert

2.Display

3.Search

4.Sort

5.Delete

6.Exit

Enter your choice:1

Enter Item Name:pen

Enter Item Quantity:10

Enter Item Cost:120

Enter Item Code:1003

\*\*\* Menu \*\*\*

1.Insert

2.Display

3.Search

4.Sort

5.Delete

6.Exit

Enter your choice:2

Item Name:pen

Item Quantity:10

Item Cost:120

Item Code:1003

\*\*\* Menu \*\*\*

1.Insert

2.Display

3.Search

4.Sort

5.Delete

6.Exit

Enter your choice:3

Enter Item Code to search:1003

Found.

Item Name : pen

Item Quantity : 10

Item Cost : 120

Item Code: 1003

\*\*\* Menu \*\*\*

1.Insert

2.Display

3.Search

4.Sort

5.Delete

6.Exit

Enter your choice:4

Sorted on Cost

Item Name:pen

Item Quantity:10

Item Cost:120

Item Code:1003

\*\*\* Menu \*\*\*

1.Insert

2.Display

3.Search

4.Sort

5.Delete

6.Exit

Enter your choice:5

Enter Item Code to delete:1003

Deleted.

\*\*\* Menu \*\*\*

1.Insert

2.Display

3.Search

4.Sort

5.Delete

6.Exit

Enter your choice:6

**Assignment No.7**

**Program:**

#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:**

========Population of states in India==========

Size of populationMap5

Enter name of the state :Maharashtraa

Maharashtra 's population is 7026357