**Assignment 1**

**Implement a class Complex which represents the Complex Number data type. Implement the following operations:**

**1. Constructor (including a default constructor which creates the complex number 0+0i).**

**2. Overloaded operator+ to add two complex numbers.**

**3. Overloaded operator\* to multiply two complex numbers.**

**4. Overloaded << and >> to print and read Complex Numbers**

#include<iostream>

using namespace std;

class complex

{

private:

int x,y;

public:

complex() //default constructor

{

x=0;

y=0;

}

complex(int real,int img) //parameterized constructor

{

x=real;

y=img;

}

complex operator+(complex c) //+ operator overloading

{

complex temp;

temp.x=x+c.x;

temp.y=y+c.y;

return temp;

}

complex operator\*(complex c) //\* operator overloading

{

complex temp;

temp.x=(x\*c.x)-(y\*c.y);

temp.y=(y\*c.x)+(x\*c.y);

return temp;

}

friend istream &operator>>(istream &,complex &); //friend function with >>,<< operator overloading

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

};

istream &operator>>(istream &a,complex &c)

{

a>>c.x>>c.y;

return a;

}

ostream &operator<<(ostream &k,complex &c)

{

k<<c.x<<"+"<<c.y<<"i";

return k;

}

int main()

{

complex c1; //default constructor with 0+0i

cout<<c1<<endl;

complex c2(2,5); //parameterized constructor

cout<<c2<<endl;

complex c3; //take user defined 1st complex number

cout<<"enter the first comp no";

cin>>c3;

complex c4; //take user defined 2nd complex number

cout<<"\nenter the sec comp no";

cin>>c4;

complex c5;

cout <<"\nadd is";

c5=c3+c4; //Call +operator overloading

cout<<c5<<endl;

cout<<"mul is";

c5=c3\*c4; //Call \*operator overloading

cout<<c5<<endl;

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Output \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

0+0i

2+5i

enter the first comp no : 1 //Enter 1 and 2

2

enter the sec comp no :3 //Enter 3 and 4

4

add is 4+6i

mul is -5+10i

**Assignment 2**

#include<iostream>

#include<string.h>

#include<stdlib.h>

using namespace std;

class Student

{

private:

char \*name,\*class1,\*div,\*dob,\*bldgrp, \*address,\*drv;

int rno,phone;

static int counter;

public:

Student();

Student(char \*,int,char \*,int,char \*,char \*,char \*,char \*,char \*);

~Student();

inline void getDetails();

inline void showDetails();

static void count();

};

int Student::counter;

Student::Student()

{

name=new char[1];

class1=new char[1];

div=new char[1];

dob=new char[1];

bldgrp=new char[1];

address=new char[1];

drv=new char[1];

rno=0;

phone=0;

}

Student::Student(char \*name,int rno, char \*address,int phone,char \*class1,char \*div,char \*dob,char \*bldgrp,char \*drv)

{

int len=strlen(name);

this->name=new char[len+1];

strcpy(this->name,name);

len=strlen(class1);

this->class1=new char[len+1];

strcpy(this->class1,class1);

len=strlen(div);

this->div=new char[len+1];

strcpy(this->div,div);

len=strlen(dob);

this->dob=new char[len+1];

strcpy(this->dob,dob);

len=strlen(bldgrp);

this->bldgrp=new char[len+1];

strcpy(this->bldgrp,bldgrp);

len=strlen(address);

this->address=new char[len+1];

strcpy(this->address,address);

len=strlen(drv);

this->drv=new char[len+1];

strcpy(this->drv,drv);

this->rno=rno;

this->phone=0;

}

Student::~Student()

{

delete name;

delete address;

delete dob;

delete drv;

delete bldgrp;

delete class1;

delete div;

rno=0;

phone=0;

cout<<"\nThank you for deleting details\n";

}

void Student::getDetails()

{

cout<<"\n\*\*\*\*\*\*Enter Details of Student \*\*\*\*\*\*\*\*\n";

cout<<"Name : ";

cin>>name;

cout<<"Roll No. : ";

cin>>rno;

cout<<"Class : ";

cin>>class1;

cout<<"Div : ";

cin>>div;

cout<<"Address: ";

cin>>address;

cout<<"Phone : ";

cin>>phone;

cout<<"Date of Birth : ";

cin>>dob;

cout<<"Driving License : ";

cin>>drv;

cout<<"Blood Group : ";

cin>>bldgrp;

}

void Student::showDetails()

{

Student::count();

cout<<"\nName : "<<name;

cout<<"\tRoll No. : "<<rno;

cout<<"\nClass : "<<class1;

cout<<"\tDiv : "<<div;

cout<<"\nAddress: "<<address;

cout<<"\tPhone : "<<phone;

cout<<"\nDate of Birth : "<<dob;

cout<<"\tDriving License : "<<drv;

cout<<"\tBlood Group : "<<bldgrp<<endl;

}

void Student::count()

{

counter++;

cout<<"\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Details of Student "<<counter<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

}

int main()

{

cout<<"\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Student Information System\*\*\*\*\*\*\*\*\*\*\*\n";

cout<<"How many student you have?";

int num;

cin>>num;

Student s[num];

for(int i=0;i<num;i++)

{

s[i].getDetails();

}

system("clear");

cout<<"\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Student Information System\*\*\*\*\*\*\*\*\*\*\*\n";

for(int i=0;i<num;i++)

{

s[i].showDetails();

}

Student s1("ABC",5,"Pune",12345,"SE","A","10-07-2016","B+","777");

s1.showDetails();

return 0;

}

Output :

How many student you have?2

\*\*\*\*\*\*\*Enter Details of Student \*\*\*\*\*\*\*\*

Name : rohit

Roll No. : 44

Class : se

Div : a

Address: wagholi

Phone : 12345678

Date of Birth : 9

Driving License : 007

Blood Group : a+

\*\*\*\*\*\*\*Enter Details of Student \*\*\*\*\*\*\*\*

Name : abhishek

Roll No. : 12

Class : se

Div : a

Address: vadgaonsheri

Phone : 98765432

Date of Birth : 7

Driving License : 700

Blood Group : b+

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Student Information System\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Details of Student 1\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Name : rohit Roll No. : 44

Class : se Div : a

Address: wagholi Phone : 12345678

Date of Birth : 9 Driving License : 007 Blood Group : a+

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Details of Student 2\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Name : abhishek Roll No. : 12

Class : se Div : a

Address: vadgaonsheri Phone : 98765432

Date of Birth : 7 Driving License : 700 Blood Group : b+

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Details of Student 3\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Name : ABC Roll No. : 5

Class : SE Div : A

Address: Pune Phone : 0

Date of Birth : 10-07-2016 Driving License : 777 Blood Group : B+

Thank you for deleting details

Thank you for deleting details

Thank you for deleting details

**Assignment 3**

**Imagine a publishing company which does marketing for book and audiocassette versions. Create a class publication that stores the title (a string) and price (type float) of a publication. From this class derive two classes: book, which adds a page count (type int), and tape, which adds a playing time in minutes (type float). Write a program that instantiates the book and tape classes, allows user to enter data and displays the data members. If an exception is caught, replace all the data member values with zero values.**

#include<iostream>

#include<cstring>

#include<exception>

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 books : public publication

{

int pages;

public:

books()

{

int pagecount=0;

}

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

{

int pagecount=pc;

}

void getb();

void displayb();

};

class tape : public publication

{

float min;

public:

void CD()

{

float time=0.0;

}

void CD(string t,float p,float tim)

{

time=tim;

}

void gett();

void displayt();

};

void books::getb()

{

cout<<"Enter the datails of the book :"<<endl;

cout<<"Enter the title of the book :"<<flush;

cin.ignore(1,'\n');

getline(cin,title);

cout<<"Enter the price of the book :"<<flush;

cin>>price;

cout<<"Enter the pages of the book :"<<flush;

cin>>pages;

try

{

if(pages > 500 && pages <1500)

{

if(price >100 && price < 2000)

{

displayb();

}

}

else throw(pages);

}

catch(int i)

{

cout<<"Caught exception in function of book"<<endl;

cout<<"You entered an invalid data"<<endl;

title = "0";

price = 0.0;

pages = 0;

displayb();

throw;

}

}

void books::displayb()

{

cout<<"The book you entered is :"<<endl<<endl;

cout<<"The entered title is :"<<title<<endl;

cout<<"The entered price is :"<<price<<endl;

cout<<"The entered pages are :"<<pages<<endl;

}

void tape::gett()

{

cout<<"Enter the details of the tape :"<<endl;

cout<<"Enter the title of the tape :"<<flush;

cin.ignore(1,'\n');

getline(cin,title);

cout<<"Enter the price of the tape :"<<flush;

cin>>price;

cout<<"Enter the pages of the playing time(in minutes):"<<flush;

cin>>min;

try

{

if(min > 30.00 && min < 90.00)

{

if(price > 500 && price < 1000)

{

displayt();

}

}

else throw(min);

}

catch(float f)

{

cout<<"caught exception in the function of tape"<<endl;

cout<<"you entered an invalid data"<<endl;

title = "0";

price = 0.0;

min = 0.0;

displayt();

throw;

}

}

void tape::displayt()

{

cout<<"The details of tape you entered is :"<<endl<<endl;

cout<<"The entered title is :"<<title<<endl;

cout<<"The entered price is :"<<price<<endl;

cout<<"The entered play time is :"<<min<<endl;

}

int main()

{

books b;

tape t;

int choice;

cout<<"Do want to buy a book(Press 1) or a tape(Press 2):"<<flush;

cin>>choice;

switch(choice)

{

case 1: try

{

b.getb();

}

catch(int i){

cout<<"Caught exception in intmain()"<<endl;

}

break;

case 2: try

{

t.gett();

}

catch(float f){

cout<<"Caught exception in intmain()"<<endl;

}

break;

default : cout<<"Entered bad choice!! Try again!!"<<endl;

}

return 0;

}

Output:

Do want to buy a book(Press 1) or a tape(Press 2) :1

Enter the details of the book :

Enter the title of the book :the man

Enter the price of the book :200

Enter the pages of the book :366

Caught exception in function of book

You entered an invalid data

The book you entered is :

The entered title is :0

The entered price is :0

The entered pages are :0

Caught exception in int main()

Do want to buy a book(Press 1) or a tape(Press 2) :2

Enter the details of the tape :

Enter the title of the tape :tape real

Enter the price of the tape :600

Enter the pages of the playing time(in minutes) :80

The details of tape you entered is :

The entered title is :tape real

The entered price is :600

The entered play time is :80

**Assignment 4**

**Write a C++ program that creates an output file, writes information to it, closes the file and open it again as an input file and read the information from the file.**

#include<iostream>

#include<stdlib.h>

#include<fstream>

using namespace std;

int main()

{

char name[30];

cout<<"Enter the name of file to create ";

cin>>name;

ofstream fout;

fout.open(name);

if(!fout)

{

cout<<"Error opening the file";

exit(1);

}

cout<<"Writing to the file"<<endl;

cout<<"Enter contents for file end it with ^D";

string data;

while(getline(cin, data))

{

if(data=="^D")

break;

fout<<data<<endl;

}

fout.close();

ifstream fin;

fin.open(name);

if(!fin)

{

cout<<"Error opening the file";

exit(1);

}

cout<<"Reading from the file\n";

while(fin)

{

getline(fin,data);

cout<<data;

}

fin.close();

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*OUTPUT\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Enter the name of file to create - Story.txt

Writing to the file

Enter contents for file end it with ^D

This is an important practical in oop

This practical includes file handling.

^D

Reading from the file

This is an important practical in oop

This practical includes file handling.

**Assignment 5**

**Write a function template selection Sort. Write a program that inputs, sorts and outputs an int array and a float array.**

#include<iostream>

using namespace std;

#define size 10

int n;

template<class T>

void selection(T a[size])

{

int min,i,j;

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<<"\n The sorted list is ....\n";

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

cout<<"\t"<<a[i];

}

int main()

{

int i,a[size];

float b[size];

cout<<"\n Selection sort";

cout<<"\n Handling integer elements";

cout<<"\n How many elements are there?";

cin>>n;

cout<<"Enter the integer numbers ";

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

cin>>a[i];

selection(a);

cout<<"\n Handling float elements";

cout<<"\n How many elements are there?";

cin>>n;

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

cin>>b[i];

selection(b);

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*OUTPUT\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Selection sort

Handling integer elements

How many elements are there? 5

Enter the integer numbers 2 5 4 3 4

The sorted list is ....

2 3 4 4 5

Handling float elements

How many elements are there? 4

1.2 4.3 2.3 5.4

The sorted list is ....

1.2 2.3 4.3 5.4

**Assignment 6**

**Write a C++ program using STL for sorting and searching user defined records such as personal records (Name, DOB, Telephone Number etc.) using vector container.**

#include <iostream> //standard input output stream header file

#include <algorithm> //The STL algorithms are generic because they can operate on a variety of data structures

#include <vector> //The header file for the STL vector library is vector.

using namespace std;

class Item //Item class is created

{

  public:

char name[10]; //varible declaration

int quantity;

int cost;

int code;

        bool operator==(const Item& i1) //Boolean operators allow you to create more complex conditional statements

{

if(code==i1.code) //operator will return 1 if the comparison is true, or 0 if the comparison is false

return 1;

return 0;

}

bool operator<(const Item& i1)

{

if(code<i1.code) //operator will return 1 if the comparison is true, or 0 if the comparison is false

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;    //if (i1.name != i2.name)

  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.";

}

}

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:mouse

Enter Item Quantity:5

Enter Item Cost:100

Enter Item Code:101

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

1.Insert 2.Display 3.Search 4.Sort 5.Delete 6.Exit

Enter your choice:1

Enter Item Name:cpu

Enter Item Quantity:5

Enter Item Cost:15000

Enter Item Code:102

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

1.Insert 2.Display 3.Search 4.Sort 5.Delete 6.Exit

Enter your choice:1

Enter Item Name:monitor

Enter Item Quantity:5

Enter Item Cost:8000

Enter Item Code:103

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

1.Insert 2.Display 3.Search 4.Sort 5.Delete 6.Exit

Enter your choice:1

Enter Item Name:keyboard

Enter Item Quantity:5

Enter Item Cost:500

Enter Item Code:104

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

1.Insert 2.Display 3.Search 4.Sort 5.Delete 6.Exit

Enter your choice:1

Enter Item Name:speaker

Enter Item Quantity:5

Enter Item Cost:1000

Enter Item Code:105

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

1.Insert 2.Display 3.Search 4.Sort 5.Delete 6.Exit

Enter your choice:2

Item Name:mouse

Item Quantity:5

Item Cost:100

Item Code:101

Item Name:cpu

Item Quantity:5

Item Cost:15000

Item Code:102

Item Name:monitor

Item Quantity:5

Item Cost:8000

Item Code:103

Item Name:keyboard

Item Quantity:5

Item Cost:500

Item Code:104

Item Name:speaker

Item Quantity:5

Item Cost:1000

Item Code:105

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

1.Insert

2.Display

3.Search

4.Sort

5.Delete

6.Exit

Enter your choice:3

Enter Item Code to search:103

Found.

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

1.Insert

2.Display

3.Search

4.Sort

5.Delete

6.Exit

Enter your choice:4

 Sorted on Cost

Item Name:mouse

Item Quantity:5

Item Cost:100

Item Code:101

Item Name:keyboard

Item Quantity:5

Item Cost:500

Item Code:104

Item Name:speaker

Item Quantity:5

Item Cost:1000

Item Code:105

Item Name:monitor

Item Quantity:5

Item Cost:8000

Item Code:103

Item Name:cpu

Item Quantity:5

Item Cost:15000

Item Code:102

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

1.Insert

2.Display

3.Search

4.Sort

5.Delete

6.Exit

Enter your choice:5

Enter Item Code to delete:105

Deleted.

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

1.Insert

2.Display

3.Search

4.Sort

5.Delete

6.Exit

Enter your choice:2

Item Name:mouse

Item Quantity:5

Item Cost:100

Item Code:101

Item Name:keyboard

Item Quantity:5

Item Cost:500

Item Code:104

Item Name:monitor

Item Quantity:5

Item Cost:8000

Item Code:103

Item Name:cpu

Item Quantity:5

Item Cost:15000

Item Code:102

**Assignment 7**

**Write a program in C++ to use map associative container. The keys will be the names of states and the values will be the populations of the states. When the program runs, the user is prompted to type the name of a state. The program then looks in the map, using the state name as an index and returns the population of the state.**

#include <iostream>

#include <map>

#include <string>

#include <utility>

using namespace std;

int main()

{

typedef map<string,int> mapType;

mapType populationMap;

populationMap.insert(pair<string, float>("Maharashtra", 125));

populationMap.insert(pair<string, float>("Uttar Pradesh", 225));

populationMap.insert(mapType::value\_type("Bihar", 120));

populationMap.insert(mapType::value\_type("West Bengal", 100));

populationMap.insert(make\_pair("Madhya Pradesh", 90));

populationMap.insert(make\_pair("Tamil Nadu", 80));

populationMap.insert(make\_pair("Rajasthan", 78));

populationMap.insert(make\_pair("Andhra Pradesh", 53));

populationMap.insert(make\_pair("Odisha", 47));

populationMap.insert(make\_pair("Kerala", 38));

populationMap.insert(make\_pair("Telangana", 37));

populationMap.insert(make\_pair("Assam", 35));

populationMap.insert(make\_pair("Jharkhand", 38));

populationMap.insert(make\_pair("Karnataka", 68));

populationMap.insert(make\_pair("Gujarat", 70));

populationMap.insert(make\_pair("Punjab", 31));

populationMap.insert(make\_pair("Chhattisgarh", 30));

populationMap.insert(make\_pair("Haryana", 29));

populationMap.insert(make\_pair("UT Delhi", 19));

populationMap.insert(make\_pair("UT Jammu and Kashmir", 14));

populationMap.insert(make\_pair("Uttarakhand", 12));

populationMap.insert(make\_pair("Himachal Pradesh", 8));

mapType::iterator iter = --populationMap.end();

populationMap.erase(iter);

cout << "Total state and UT of India with Size of populationMap: " << populationMap.size() << '\n';

for (iter = populationMap.begin(); iter != populationMap.end(); ++iter)

{

 cout << iter->first <<":" << iter->second << " million\n";

}

char c;

do

{

string state;

cout<<"\nEnter that state you want to know the population of: ";

cin>>state;

iter = populationMap.find(state);

if( iter != populationMap.end() )

    cout << state <<"'s populations is "

        << iter->second << " million\n";

else

    cout << "State is not in populationMap" << '\n';

cout<<"Do you wish to continue?(y/n):";

cin>>c;

}while(c=='y'||c=='Y');

populationMap.clear();

return 0;

}

OUTPUT-

Enter that state you want to know the population of: Assam

Assam's populations is 35 million

Do you wish to continue?(y/n):n