# 1) Implement a class Complex which represents the Complex Number data type. Implement Constructor , Overload operator

#include <iostream>

using namespace std;

class Complex {

private:

 int real, imag;

public:

 //constructor

 Complex(int r = 0, int i = 0) {real = r; imag = i;}

 //friend functions of complex class

 friend ostream & operator << (ostream &out, Complex const &obj);

 friend istream & operator >> (istream &in, Complex &obj);

 //overloading of + operator

 Complex operator + (Complex const &obj)

 {

 Complex res;

 res.real = real + obj.real;

 res.imag = real + obj.imag;

 return res;

 }

 //overloading of \* operator

 Complex operator \* (Complex const &obj)

 {

 Complex res;

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

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

 return res;

 }

};

ostream & operator << (ostream &out, Complex const &obj)

{

 out << obj.real;

 out << "+i" << obj.imag;

 return out;

}

istream & operator >> (istream &in, Complex &obj)

{

 cout << "\n Enter Real Part: "<<endl;

 in >> obj.real;

 cout << "\n Enter Imag Part: "<<endl;

 in >> obj.imag;

 return in;

}

int main()

{

 Complex c1,c2,c3,c4;

 cout <<"\n Enter First Number : ";

 cin >> c1;

 cout <<"\n Enter Second Number : ";

 cin >> c2;

 //Addition

 c3 = c1+c2;

 cout << "\n The Addition Is : " <<c3 ;

 //Multiplication

 c4 = c1 \* c2;

 cout << "\n The Multiplication Is :" <<c4 <<endl;

 return 0;

}

# 2) Develop a program in C++ to create a database of student’s information system containing the following information: Name, Roll number, Class, Division, Date of Birth, Blood group, Contact address, Telephone number, Driving license no. and other. Construct the database with suitable member functions. Make use of constructor, default constructor, copy constructor, destructor, static member functions, friend class, this pointer, inline code and dynamic memory allocation operators-new and delete as well as exception handling.

#include<iostream>

#include<string>

using namespace std;

class student

{

private:

 string name,dob,bloodgrp,addr,yearclass,mob;

 int div;

 int \*rollno;

public:

 friend class faculty;

 student()

 { rollno = new int;

 \*rollno = div =-1;

 name = bloodgrp=dob=addr=yearclass=mob="";

 }

~student()

{

delete rollno; //deallocation

}

void adddata()

{

 cout<<"\n enter the student information:";

 cin.ignore();

 getline(cin,name);

 cout<<"enter the Rollno: ";

 cin>>\*rollno;

 cout<< "enter divison (1/2/3/4/5):";

 cin>>div;

 cout<<"enter the birth date: "<<endl;

 cin>>dob;

 cout<<"enter the blood group: ";

 cin>>bloodgrp;

 cout<<"enter the phone number: ";

 cin>>mob;

 cout<<"enter the address: ";

 cin>>addr;

}

void display()

{

 cout<<"\n-----------------------------------";

 cout<<"\n Name :"<<name;

 cout<<"\nRoll No :"<< \*rollno;

 cout<<"\nYear(SE/TE/BE):"<<yearclass;

 cout<<"\nDiv(1/2/3/4/5):"<<div;

 cout<<"\nBlood Group :"<<dob;

 cout<<"\nMobile Number :"<<bloodgrp;

 cout<<"\nAddress :"<<addr;

 cout<<"\n------------------------------------";

}

 static void header()

 {

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

 }

};

class faculty

{

private:

 int id;

public:

 faculty()

 {

 id = 000;

 }

 faculty(const faculty &f1)

 {

 id = f1.id;

 }

 void fdisplay(student &obj,int f\_d)

 {

 try

 {

 if (obj.div == f\_d)

 obj.display();

 else

 throw( obj.div);

 }

 catch( int x)

 {

 cout<<"\n Invalid Divison....";

 }

 }

};

int main() {

 student st[5];

 faculty f;

 int ch=0,count=0;

 do

 {

 cout<<"\n \* \* \* Student information system \*";

 cout<<"\n \* \* \* Menu \* \* \*";

 cout<<"\n 1.Add Information ";

 cout<<"\n 2. Display Information ";

 cout<<"\n 3. Faculty wise Information";

 cout<<"\n 4. Exit ";

 cout<<"\n Enter choice : ";

 cin>>ch;

 switch(ch)

 {

 case 1: st[count].adddata();

 count++;

 break;

 case 2:

 for (int j=0;j<count;j++)

 {

 student::header();

 st[j].display();

 }

 break;

 case 3:

 int f\_div;

 cout<<"\n enter Divison of Faculty: ";

 cin>> f\_div;

 for ( int j=0;j<count;j++)

 {

 cout<<"\n \* \* \* Student Information System (Faculty RAS) \* \* \*";

 f.fdisplay(st[j],f\_div);

 }

 break;

 case 4:

 exit(0);

 }

 }while ( ch!= 4);

 return 0;

 }

# 3) Imagine a publishing company which does marketing for book and audio cassette versions. Create a class publication that stores the title (a string) and price (type float) of publications. 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 class, 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>

#program by Narendra Dwivedi Intel 20U520

#include<string>

using namespace std;

//base class publication

class publication

{

private:

string title;

float prices;

public:

publication()

 {

title="";

prices=0.0;

}

void get\_data()

{

cout<<"\nEnter Title :";

cin.ignore();//clear input buffer

getline(cin,title);

cout<<"\nEnter Price : ";

cin>>prices;

}

void put\_data()

{

cout<<"\n \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \n";

cout<<"\n Information : " <<endl;

cout<<"\n Title :"<<title;

cout<<"\n Price :"<<prices;

}

};

class book: public publication

{

private:

int pages;

public:

book(){

pages=0;

}

void get\_data()

{

publication::get\_data();

cout<<endl;

cout<<"Enter Page Count : \n";

cin>>pages;

}

void put\_data()

{

publication::put\_data();

try{

if(pages<0)

 throw pages;}

 catch(int f)

 {

 cout<<"\n error: pages not valid :"<<f;

 pages=0;

 }

cout<<"\n Pages Are :"<<pages;

}

};

class tape: public publication

{

private:

float playtime;

public:

tape()

{

playtime=0.0;

}

void get\_data()

{

publication::get\_data();

cout<<"Enter Play Time Of Cassette \n";

cin>>playtime;

}

void put\_data()

{

publication::put\_data();

try

{

if(playtime<0.0)

throw playtime;

}

catch(float r)

{

cout<<"\n Error: Invalid Playtime : "<<playtime;

playtime=0.0;

}

cout<<"\n Playtime is : "<<playtime;

}

};

int main()//main program

{

book b[10];// arrray of objects

tape t[10];

int choice=0,bookCount=0,tapeCount=0;

cout<<"-----------------------";

do

{

cout<<"\n 1. Add book ";

cout<<"\n 2. Add tape: ";

cout<<"\n 3. Display book ";

cout<<"\n 4. Display tape";

cout<<"\n 5. Exit:"<<endl;

cout<<"\n Enter Choice : ";

cin>>choice;

switch(choice)

{

case 1:

 {

 cout<<"\n--------------\n";

 cout<<"Add Book: \n";

 b[bookCount].get\_data();

 bookCount++;

 break;

 }

case 2:

 {

 cout<<"\n--------------\n";

 cout<<"Add Tape: \n";

 t[tapeCount].get\_data();

 tapeCount++;

 break;

 }

case 3:

 {

 cout<<"\n (books)";

 for(int j=0;j<bookCount;j++)

 {

 b[j].put\_data();

 }

 break;

 }

case 4:

 {

 cout<<"\n (tape)";

 for(int j=0;j<tapeCount;j++)

 {

 t[j].put\_data();

 }

 break;

 }

case 5:

 {

 cout<<"\*\*\*\*\*\*\*\*\*\*Program Exited

Successfully\*\*\*\*\*\*\*\*\*\*"<<endl;

exit(0);

 }

 default:

 {

 cout<<"\n Invalid";

 }

}

}

while(choice!=5);

return 0;

}

# 4)write c++ class to create a output file,write information to it,close the file open it again as an input file and read the information from the file.

# #include <iostream>

# #include <fstream>

# #include <string>

# class FileHandler {

# public:

# FileHandler(const std::string& filename) : filename(filename) {}

# // Function to write information to the file

# void writeToFile(const std::string& data) {

# std::ofstream outputFile(filename);

# if (outputFile.is\_open()) {

# outputFile << data;

# outputFile.close();

# std::cout << "Data written to file successfully." << std::endl;

# } else {

# std::cerr << "Unable to open the file for writing." << std::endl;

# }

# }

# // Function to read information from the file

# void readFromFile() {

# std::ifstream inputFile(filename);

# if (inputFile.is\_open()) {

# std::string line;

# std::cout << "Reading data from the file:" << std::endl;

# while (std::getline(inputFile, line)) {

# std::cout << line << std::endl;

# }

# inputFile.close();

# } else {

# std::cerr << "Unable to open the file for reading." << std::endl;

# }

# }

# private:

# std::string filename;

# };

# int main() {

# // Specify the filename

# std::string filename = "example.txt";

# // Create an instance of the FileHandler class

# FileHandler fileHandler(filename);

# // Write data to the file

# fileHandler.writeToFile("Hello, this is some information written to the file.");

# // Read data from the file

# fileHandler.readFromFile();

# return 0;

# }

# 5) Write a function template for selection sort that inputs, sorts and outputs an integer array and a float array.

#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 choice;

 char C[size];

 int A[size];

 float B[size];

 int i;

 cout<<"-----------------------";

 do

 {

 cout<<"\n 1. Integer : ";

 cout<<"\n 2. Float : ";

 cout<<"\n 3. Exit : "<<endl;

 cout<<"\n Enter Choice : ";

 cin>>choice;

 switch(choice)

 {

 case 1:

 {

 cout<<"\nEnter Total Number Of Integer Elements:";

 cin>>n;

 cout<<"\nEnter Integer Elements:";

 for(i=0;i<n;i++)

 {

 cin>>A[i];

 }

 sel(A);

 break;

 }

 case 2:

 {

 cout<<"\nEnter Total Number Of Float Elements:";

 cin>>n;

 cout<<"\nEnter Float Elememts:";

 for(i=0;i<n;i++)

 {

 cin>>B[i];

 }

 sel(B);

 break;

 }

 case 3:

 {

 cout<<"\*\*\*\*\*\*\*\*\*\*Program Exited Successfully\*\*\*\*\*\*\*\*\*\*"<<endl;

 exit(0);

 }

 default:

 {

 cout<<"\n Invalid";

 }

 }

 }

 while(choice!=4);

return 0;

}

# 6) Write C++ program using STL for sorting and searching user defined records such as Item records (Item code, name, cost, quantity etc) using vector container.

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.";

 }

}

# 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>

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();

}