1.) Write a C++ program to implement stack using following

functions:

Push

Pop

Display

#include<iostream>

using namespace std;

#define size 5

class stack

{

int stck[size];

int tos;

public:

void init();

void push();

int pop();

void display();

};

void stack::init()

{

tos=0;

}

void stack::push()

{

int i;

if(tos==size)

{

cout<<"stack overflow \n";

return;

}

cout<<"enter element \n";

cin>>i;

stck[tos]=i;

tos++;

}

int stack::pop()

{

if(tos==0)

{

cout<<"stack underflow \n";

return 1;

}

tos--;

cout<<"popped element is"<<stck[tos];

return 0;

}

void stack::display()

{

if(tos==0)

{

cout<<"stack empty \n";

return;

}

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

{

cout<<stck[i]<<"\n";

}

}

int main()

{

int ch,i;

stack s;

s.init();

while(1)

{

cout<<"\n1.push 2.pop 3.display \n";

cin>>ch;

switch(ch)

{

case 1:s.push();

break;

case 2:s.pop();

break;

case 3:s.display();

break;

default: return 0;

}

}

}

2(a)Write a C++ program to read the data of n employee and compute net salary of each employee using pointer. Given that an employee class contains following :- Data members: Employee no, Employee name, Basic salary, DA, IT, Net salary, gross salary Member functions: To read data, to calculate net salary and to print data [DA = 52% of basic salary, IT = 30% of gross salary, Gross salary = DA + Basic, Net salary = DA + Basic – IT].

#include<iostream>

using namespace std;

class employee

{

int num, basic;

long da,it,netsal,gsal;

char name[20];

public:

void read();

void disp();

void calc();

};

void employee::read()

{

cout<<"enter employee id,name & basic salary \n";

cin>>num>>name>>basic;

}

void employee::calc()

{

da=(0.52)\*basic;

gsal=da+basic;

it=(0.3)\*gsal;

netsal=basic+da-it;

}

void employee::disp()

{

cout<<num<<"\t"<<name<<"\t\t\t"<<basic<<"\t\t"<<da<<"\t\t"<<it<<"\t\t"

<<gsal<<"\t"<<netsal<<"\n";

}

int main()

{

int n;

cout<<"enter no. of employee \n";

cin>>n;

employee e[n],\*p;

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

{

p=&e[i];

p->read();

p->calc();

}

cout<<"sl no. eid\t name \t\t basic salary\t \t DA\t\tgross salary\tincome tax\t Net salary\n";

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

{

p=&e[i];

p->disp();

}

return 0;

}

2) (b) Write a C++ program to find the largest of three numbers using inline function and default argument concept.

#include<iostream>

using namespace std;

inline float lar(float a, float b ,float c=50 )

{

return ((a>b&&a>c)?a:b>c?b:c);

}

int main()

{

float x,y,z;

cout<<"\nEnter three numbers:\n";

cin>>x>>y>>z;

cout<<"Largest= "<<lar(x,y,z);

cout<<"\nTaking 3rd number as 50\n";

cout<<"\nEnter two numbers:\n";

cin>>x>>y;

cout<<"\nLargest using default value= "<<lar(x,y)<<"\n";

return 0;

}

3.) Write a C++ program to define a student class with data members USN, name and marks of 3 subjects. And member functions to read, display, and to calculate average of best 2 marks. Also find who is the topper among “n” no. of students.

#include<iostream>

using namespace std;

class stud

{

char usn[20],name[30];

float marks[3];

int i;

public:

float avg;

void read();

void calc(int);

void disp();

};

void stud::read()

{

cout<<"\nEnter USN no.: ";

cin>>usn;

cout<<"\nEnter name: ";

cin>>name;

for(i=0;i<3;i++)

{

cout<<"\nEnter the marks of subject "<<i+1<<": ";

cin>>marks[i];

}

}

void stud::calc(int n)

{

int sum=0,min=marks[0];

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

{

sum+=marks[i];

if(min>=marks[i])

min=marks[i];

}

avg=float(sum-min)/2;

}

void stud::disp()

{

cout<<usn<<"\t"<<name<<"\t";

for(i=0;i<3;i++)

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

cout<<avg<<"\n";

}

int main()

{

int n,topper,i,z;

float max=0;

cout<<"\nEnter the number of students: ";

cin>>n;

stud s[n];

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

{

s[i].read();

s[i].calc(n);

}

cout<<"USN\tName\t";

for(i=0;i<3;i++)

{

cout<<"Marks "<<i+1<<"\t";

}

cout<<"Avg.\n";

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

s[i].disp();

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

{

if(max<s[i].avg)

{

max=s[i].avg;

topper=i;

}

}

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

{

if(s[topper].avg==s[i].avg)

{

cout<<"\nThe toppers is student "<<i+1<<": ";

cout<<"\nDetails:-\n";

s[i].disp();

}

}

}

4.a) Design a C++ program to implement access control to some shared resource used by all objects of a class using a static variable.

#include<iostream>

using namespace std;

class resource

{

static int res;

public:

static int getr();

void free\_res()

{

res=0;

}

};

int resource::res;

int resource::getr()

{

if(res)

return 0;

else

{

}

}

res=1;

return 1;

int main()

{

resource a,b;

if(resource::getr())

cout<<"Resource under use, object a is using \n";

if(!resource::getr())

cout<<"Resource busy, object b access denied \n";

a.free\_res();

if(resource::getr())

cout<<"Resource can now be used by Object b \n";

return 0;

}

4.b) Design a C++ program to keep the track of the number of objects of a particular class type that are inexistence using a static variable.

#include <iostream>

using namespace std;

class Counter

{

public:

static int count;

Counter() { count++; }

~Counter() { count--; }

};

int Counter::count;

void f();

int main(void)

{

Counter o1;

cout << "Objects in existence: ";

cout << Counter::count << "\n";

Counter o2;

cout << "Objects in existence: ";

cout << Counter::count << "\n";

f();

cout << "Objects in existence: ";

cout << Counter::count << "\n";

return 0;

}

void f()

{

Counter temp;

cout << "Objects in existence: ";

cout << Counter::count << "\n";

// temp is destroyed when f() returns

}

5) Design a C++ program to implement a class which accepts date in different formats (using constructor overloading).

#include <iostream>

#include <cstdio>

using namespace std;

class date

{

int day, month, year;

public:

date(char \*d);

date(int m, int d, int y);

void show\_date();

};

// Initialize using string.

date::date(char \*d)

{

sscanf(d, "%d%\*c%d%\*c%d", &month, &day, &year);

}

// Initialize using integers.

date::date(int m, int d, int y)

{

day = d;

month = m;

year = y;

}

void date::show\_date()

{

cout << month << "/" << day;

cout << "/" << year << "\n";

}

int main()

{

date ob1(12, 4, 2003), ob2("10/22/2003");

ob1.show\_date();

ob2.show\_date();

return 0;

}

6. Design a C++ program to create class called list with member functions to insert an element from front as well as to delete element from front of list. Demonstrate all functions by creating list object.

#include<iostream>

#include<new>

using namespace std;

struct nod

{

int info;

struct nod\*next;

};

typedef struct nod node;

class list

{

node \*f;

public:

list()

{

f=NULL;

}

void ins(int num)

{

node \*p=new node;

p->info=num;

p->next=f;

f=p;

}

void del()

{

node \*temp=f;

if(f==NULL)

cout<<"\nNo elements to delete.\n";

else

{

}

cout<<"\n The deleted elements is :\n"<<f->info;

f=f->next;

delete temp;

cout<<"\n Deletion successfull \n";

return;

}

void disp()

{

node \*temp=f;

if(f==NULL)

cout<<"\n List is empty \n";

else

{

cout<<"\n Elements in the list are: ";

while(temp!=NULL)

{

cout<<" "<<temp->info;

temp=temp->next;

}

}

}

};

int main()

{

int num,ch=1;

list ob;

cout<<"\n!!!!!!!!!!! LINEAR LINK LIST !!!!!!!!!!!!!!!!\n";

cout<<"\n1] Insert\n2] Delete\n3] Exit";

while(ch)

{

cout<<"\nEnter your choice \n";

cin>>ch;

switch(ch)

{

case 1: cout<<"\n Enter no. to be insrted\n";

cin>>num;

ob.ins(num);

ob.disp();

break;

case 2: ob.del();

ob.disp();

break;

case 3: return 0;

default:cout<<"Invalid choice \n";

break;

}

}

}

7. Design a C++ program for a hospital to create a database regarding its indoor patients. (Identify the member function). create a base class to store above information, member function should include functions to enter information and display list of all the patients in the database. Create a derived class to store the information about paediatric patients (less than 12yrs age).

#include <iostream>

using namespace std;

struct date

{

int d;

int m;

int y;

};

class hospital

{

char name[100];

struct date d\_adm;

struct date d\_dis;

protected:int age;

public:

void getdata()

{

cout<<"Enter name of the patient: ";

cin>>name;

cout<<"Enter age: ";

cin>>age;

cout<<"Enter date of admission: ";

cin>>d\_adm.d>>d\_adm.m>>d\_adm.y;

cout<<"Enter date of discharge: ";

cin>>d\_dis.d>>d\_dis.m>>d\_dis.y;

}

void display()

{

cout<<name<<"\t";

cout<<age<<"\t";

cout<<d\_adm.d<<'-'<<d\_adm.m<<'-'<<d\_adm.y<<"\t \t";

cout<<d\_dis.d<<'-'<<d\_dis.m<<'-'<<d\_dis.y;

}

};

class pediatric\_patient:public hospital

{

char vaccine[20];

public:

void get()

{

getdata();

cout<<"enter the name of vaccine to be given \n";

cin>>vaccine;

}

void put()

{

if(age<12)

{

display();

cout<<"\t"<<"\t";

cout<<vaccine;

cout<<"\n";

}

else

}

};

cout<<"age greater than 12 not a pediatric patient";

int main()

{

hospital h[5];

int n;

cout<<"Enter the number of patients \n";

cin>>n;

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

{

h[i].getdata();

}

cout<<"Patient database \n";

cout<<"NAME" <<"\t" <<"AGE" <<"\t" <<"DATE\_OF\_ADMISSION "<<"\t"

<<"DATE\_OF\_DISCHARGE \n";

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

{

h[i].display();

cout<<"\n";

}

pediatric\_patient a1[5];

cout<<"Enter the number of pediatric patients \n";

cin>>n;

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

{

a1[i].get();

}

cout<<"pediatric Patient database \n";

cout<<"NAME" <<"\t" <<"AGE" <<"\t" <<"DATE\_OF\_ADMISSION "<<"\t"

<<"DATE\_OF\_DISCHARGE"<<"\t"<<"VACCINE \n";

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

{

a1[i].put();

}

return 0;

}

8. Design a C++ program for exception handling. Create a user defined exception classes for divide by zero and negative number input separately.

#include <iostream>

#include <cstring>

using namespace std;

class Negative\_number

{

public:

char str\_what[80];

int what;

Negative\_number(char \*s, int e)

{

strcpy(str\_what, s);

what = e;

}

};

class Divide\_by\_zero

{

public:

char str\_what[80];

int what;

Divide\_by\_zero(char \*s, int e)

{

strcpy(str\_what, s);

what = e;

}

};

int main()

{

int i,nr,dr;

try

{

cout << "Enter a positive number: ";

cin >> i;

if(i<0)

throw Negative\_number("Not Positive", i);

}

catch (Negative\_number e)

{ // catch an error

cout << e.str\_what << ": ";

cout << e.what << "\n";

}

try

{

cout << "Enter the value of numerator: ";

cin >> nr;

cout << "Enter the value of denominator: ";

cin >> dr;

if(dr==0)

throw Divide\_by\_zero("Cannot divide by zero", dr);

}

catch (Divide\_by\_zero e)

{ // catch an error

cout << e.str\_what << ": ";

cout << e.what << "\n";

}

return 0;

}

9. Design a C++ program for sorting names using file handling.

#include <iostream>

#include <fstream>

using namespace std;

int main()

{

ifstream in("C:\\file1.txt"); // input

if(!in) {

cout << "Cannot open input file.\n";

return 1;

}

ofstream out("TextFile1.txt"); // output

if(!out) {

cout << "Cannot open output file.\n";

return 1;

}

char str[5][20];

char t[20];

int i, j;

cout<<"file contents \n";

for(int i=0;i<5;i++)

{

in>>str[i];

cout<<str[i]<<endl;

}

for(i=1; i<5; i++)

{

for(j=1; j<5; j++)

{

if(strcmp(str[j-1], str[j])>0)

{

strcpy(t, str[j-1]);

strcpy(str[j-1], str[j]);

strcpy(str[j], t);

}

}

}

cout<<"Strings (Names) in alphabetical order : \n";

for(i=0; i<5; i++)

{

cout<<str[i]<<"\n";

out<<str[i]<<"\n";

}

}

10. Design a C++ program to implement following operator overloading concept using complex number. + , - ,+ +, --, = =.

#include <iostream>

using namespace std;

class complex

{

int real, imag;

public:

complex() {}

complex(int r, int i) {

real = r;

imag = i;

}

void show() {

cout << real << " ";

cout << imag << "\n";

}

complex operator+(complex op2);

complex operator-(complex op2);

complex operator++();

complex operator--();

bool operator==(complex op2);

};

complex complex::operator+(complex op2)

{

complex temp;

temp.real = op2.real + real;

temp.imag = op2.imag + imag;

return temp;

}

complex complex::operator-(complex op2)

{

complex temp;

temp.real = real - op2.real;

temp.imag = imag - op2.imag;

return temp;

}

complex complex::operator++()

{

real++;

imag++;

return \*this;

}

complex complex::operator--()

{

real--;

imag--;

return \*this;

}

bool complex::operator==(complex op2)

{

bool result = (op2.real == real) && (op2.imag == imag);

return result;

}

int main()

{

complex comp1(10, 20), comp2( 5, 30);

complex comp3;

comp1.show();

comp2.show();

cout<<"Addition of complex numbers:";

comp3 = comp1 + comp2;

comp3.show();

cout<<"Subtraction of complex numbers:";

comp3 = comp1 - comp2;

comp3.show();

cout<<"Checking if the 2 complex numbers are equal\n";

comp1.show();

comp2.show();

bool result = comp1==comp2;

if(result)

cout<<"Numbers are same\n";

else

cout<<"Numbers are different\n";

cout<<"Incrementing complex number:";

comp1.show();

comp1++;

cout<<"After increment:";

comp1.show();

cout<<"Decrementing complex number:";

comp2.show();

comp2--;

cout<<"After decrement:";

comp2.show();

return 0;

}

11. Design a C++ program to convert dollar to rupees, euro to rupees and pound to rupees using pure virtual functions. 1 dollar = 54.3 Rs, 1 pound = 81.1Rs, 1 euro = 70Rs.

#include<iostream>

using namespace std;

class rs

{

public:

float rupees;

virtual void conv()=0;

void disp()

{

cout<<" is eqvivalent to "<<rupees<<" INR \n";

}

};

class doll:public rs

{

float dol;

public:

void conv()

{

cout<<"Enter currncy in dollar \n";

cin>>dol;

rupees=54.3\*dol;

cout<<" "<<dol<<" in dollar ";

disp();

}

};

class euro:public rs

{

float er;

public:

void conv()

{

cout<<"Enter currency in Euro \n";

cin>>er;

rupees=70.2\*er;

cout<<" "<<er<<" in euro ";

disp();

}

};

class pd:public rs

{

float pnd;

public:

void conv()

{

cout<<"Enter currency in pound \n";

cin>>pnd;

rupees=81.1\*pnd;

cout<<" "<<pnd<<" in pound ";

disp();

}

};

int main()

{

int c;

doll d;

euro e;

pd p;

cout<<"\t\t\t Currency conversion \n";

while(1)

{

cout<<"1.$ to Rs. 2.Euro to Rs. 3.Pound to Rs. 4.Exit \n";

cin>>c;

switch(c)

{

case 1:d.conv();

break;

case 2:e.conv();

break;

case 3:p.conv();

break;

default:return 0;

}

}

return 0;

}

12. Design a C++ program to sort integers and floating point numbers using template.

#include <iostream>

using namespace std;

template <class X> void bubble(X \*items,int count)

{

register int a, b;

X t;

for(a=1; a<count; a++)

for(b=count-1; b>=a; b--)

if(items[b-1] > items[b])

{

}

int main()

{

// exchange elements

t = items[b-1];

items[b-1] = items[b];

items[b] = t;

}

int iarray[7] = {7, 5, 4, 3, 9, 8, 6};

double darray[5] = {4.3, 2.5, -0.9, 100.2, 3.0};

bubble(iarray, 7);

bubble(darray, 5);

cout << "Here is sorted integer array: ";

for(int i=0; i<7; i++)

cout << iarray[i] << ' ';

cout << endl;

cout << "Here is sorted double array: ";

for(int i=0; i<5; i++)

cout << darray[i] << ' ';

cout << endl;

int i;

cin>>i;

return 0;

}

13. Write a C++ program to create a file to store Account holder name, account number and Balance for given number of customers. Also retrieve the values from the file and print it on the standard output.

#include<iostream>

#include<fstream>

using namespace std;

struct account\_holder

{

char name[20];

double bal;

long acc;

};

int main()

{

int n,i;

cout<<"Enter no. of account holders \n";

cin>>n;

struct account\_holder b[10];

struct account\_holder c[10];

ofstream op("TextFile2.txt");

if(!op)

{

cout<<"Cannot open file \n";

return 1;

}

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

{

cout<<"enter a/c no.,name & bal of customer : "<<i+1<<"\n";

cin>>b[i].acc>>b[i].name>>b[i].bal;

op.write((char \*)&b[i],sizeof(struct account\_holder));

}

op.close();

ifstream ip("TextFile2.txt");

if(!ip)

{

cout<<"File doesn't exists \n";

return 1;

}

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

{

if(ip)

{

ip.read((char \*)&c[i],sizeof(struct account\_holder));

cout<<"A/c no : "<<c[i].acc<<"\n";

cout<<"Name : "<<c[i].name<<"\n";

cout<<"Balance (in INR) : "<<c[i].bal<<"\n";

}

}

ip.close();

return 0;

}

14. (a) Write a C++ program to perform ( ) overloading.

#include<iostream>

using namespace std;

class loc

{

int longitude, latitude;

public:

loc ( ) {}

loc(int lg, int lt)

{

longitude = lg;

latitude = lt;

}

void show ()

{

cout<< longitude<<" ";

cout<< latitude<<"\n ";

}

loc operator+ (loc op2);

loc operator()( int i, int j);

};

// overload () for loc

loc loc::operator()(int i, int j)

{

longitude = i;

latitude = j;

return \*this;

}

// overload + for loc

loc loc::operator+(loc op2)

{

loc temp;

temp.longitude= op2.longitude+longitude;

temp.latitude= op2.latitude+latitude;

return temp;

}

int main()

{

loc ob1(10,20), ob2(1,1);

ob1.show();

ob1(7,8);

ob1.show();

ob1=ob2+ob1(10,10);

ob1.show();

return 0;

}

14(b) Write a C++ program to perform -> overloading.

#include<iostream>

using namespace std;

class myclass{

public:

int i;

myclass \*operator->() {return this;}

};

int main()

{

myclass ob;

ob->i=10;

cout<<ob.i <<" "<<ob->i;

return 0;

}

15. Sorting using Generic classes.

#include<iostream>

#include<cstdlib>

using namespace std;

const int SIZE=10;

template<class atype>class sort

{

public: atype a[SIZE];

atype &operator[](int i)

{

if(i<0||i>SIZE-1)

{

cout<<"\nIndex value of ";

cout<<i<<" is out-of-bonds.\n";

exit(1);

}

return a[i];

}

sort(){}

void read(int z)

{

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

cin>>a[i];

}

void disp(int z)

{

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

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

cout<<"\n";

}

template<class X> void bubble(X \*items,int count)

{

register int a,b;

X t;

for(a=1;a<count;a++)

{

for(b=count-1;b>=a;b--)

{

if(items[b-1]>items[b])

{

}

}

};

int main()

{

t=items[b-1];

items[b-1]=items[b];

items[b]=t;

}

}

sort <int> intob;

sort <double> doubleob;

int i,m,n;

cout<<"\nEnter the size of integer array: -\n";

cin>>n;

cout<<"\nEnter the size of double array: -\n";

cin>>m;

intob[n-1]=0;

doubleob[m-1]=0;

cout<<"\nEnter the integer array:-\n";

intob.read(n);

cout<<"\nEnter the double array:-\n";

doubleob.read(m);

cout<<"\nUnsorted integer array is:-\n";

intob.disp(n);

cout<<"\nUnsorted double array is:-\n";

doubleob.disp(m);

intob.bubble(intob.a,n);

doubleob.bubble(doubleob.a,m);

cout<<"\nSorted integer array is:-\n";

intob.disp(n);

cout<<"\nSorted double array is:-\n";

doubleob.disp(m);

return 0;

}