C++ LAB PROGRAMS

1) Write and execute a C++ Program to display names, roll no's, and grades of 3 students who have appeared in the examination. Create a class with data members as Name, Roll no and Marks for 3 subjects. Write a method to calculate the grade. Read and display the contents of an array using pointer to array of objects

```
#include<iostream>
using namespace std;
class student
{
private:int idno;
	char name[20],grade;                    //variable declarations
	int m1,m2,m3,total,perc;
public: void input()
	{
	cout<<"Enter register number"<<endl;
	cin>>idno;
	cout<<"Enter name"<<endl;
	cin>>name;
	cout<<"Enter marks of 3 subjects"<<endl;
	cin>>m1>>m2>>m3;
	}
	void compute()
	{
	total=m1+m2+m3;          //computes total marks
	perc=total/3;            //computes percentage
	if(perc>=90)
	grade='S';
	else
	if(perc>=75)
	grade='A';
```

```
        else
        if(perc>=60)
        grade='B';
        else
        if(perc>=50)
        grade='C';

       else
       grade='F';
        }

        void display()
        {
        cout<<"Register no= "<<idno<<endl;
        cout<<"Name = "<<name<<endl;
        cout<<"Grade= "<<grade<<endl;
        }
};
main()
{
int i,n;
student s[10],*p;
p=&s[10];                                   //pointer to array of objects
cout<<"Enter the number of students\n";
cin>>n;
for(i=0;i<n;i++)
{
cout<<"Enter details of student"<<i+1<<endl;
(p+i)->input();
(p+i)->compute();
}
for(i=0;i<n;i++)
{
cout<<"Details of student"<<i+1<<endl;
```

```
(p+i)->display();
}
}
```

```
Enter the number of students
2
Enter roll number
1
Enter name
suhas
Enter total marks in 3 subjects
292
Enter roll number
2
Enter name
subramani
Enter total marks in 3 subjects
296
List of records
Student name-suhas
Roll-1
Total-98
Grade-F

Student name-suhas
Roll-1
Total-292
Grade-S

Student name-subramani
Roll-2
Total-296
Grade-S
```

2. Given that an EMPLOYEE class contains following members: data members: Employee number, Employee name, Basic, DA, IT, Net Salary .Write and execute a C++ program to read the data of N employee and compute Net salary of each employee (DA=52% of Basic and Income Tax (IT) =30% of the gross salary) using array of objects.

```
#include<iostream>
using namespace std;
class emp
```

```
{
private:int empno,basic,allowance,it,net,gross;
	char name[25];                          //variable declarations
public: void input()
	{
	cout<<"Enter employee number"<<endl;
	cin>>empno;
	cout<<"Enter name"<<endl;
	cin>>name;
	cout<<"Enter basic salary"<<endl;
	cin>>basic;
	}
	void compute()
	{
	allowance=basic*0.52;
	gross=basic+allowance;
	it=0.3*gross;
	net=gross-it;
	}
	void display()
	{
	cout<<"Name= "<<name<<endl;
	cout<<"Basic salary= "<<basic<<endl;
	cout<<"Allowance= "<<allowance<<endl;
	cout<<"Gross salary= "<<gross<<endl;
	cout<<"Income tax= "<<it<<endl;
	cout<<"Net salary= "<<net<<endl;
	}
};
main()
{
emp e[10];              //array of objects
int n,i;
cout<<"Enter the no. of employess\n";
```

```
cin>>n;
for(i=0;i<n;i++)
{
cout<<"Enter details of employee "<<i+1<<endl;
e[i].input();
e[i].compute();
}
for(i=0;i<n;i++)
{
cout<<"Details of employee "<<i+1<<endl;
e[i].display();

}

}
```

```
Enter the no. of employess
3
Enter details of employee 1
Enter employee number
1
Enter name
ramesh
Enter basic salary
22000
Enter details of employee 2
Enter employee number
2
Enter name
suresh
Enter basic salary
25000
Enter details of employee 3
Enter employee number
3
Enter name
suveer
Enter basic salary
31000
Details of employee 1
Name= ramesh
Basic salary= 22000
Allowance= 11440
Gross salary= 33440
Income tax= 10032
Net salary= 23408
Details of employee 2
Name= suresh
Basic salary= 25000
Allowance= 13000
Gross salary= 38000
Income tax= 11400
Net salary= 26600
Details of employee 3
Name= suveer
Basic salary= 31000
Allowance= 16120
Gross salary= 47120
```

3. Write and execute a C++ program to create a class called 'COMPLEX' to hold a complex number. Include a friend function to add and multiply two complex numbers.

```
#include<iostream>
using namespace std;
class comp
{
private:float real,img;
public: void input()
	{
	cout<<"Enter real and img part"<<endl;
	cin>>real>>img;
	}
	friend void add(comp c1,comp c2);		//friend functions
	friend void multi(comp c1,comp c2);

};
	void add(comp c1,comp c2)		//passing objects as parameters
	{
	float real,img;
	real=c1.real+c2.real;
	img=c1.img+c2.img;
	cout<<"Sum = "<<real<<"+"<<img<<"i"<<endl;
	}
	void multi(comp c1,comp c2)		//passing objects as parameters
	{
	float real,img;
	real=c1.real*c2.real-c1.img*c2.img;
	img=c1.real*c2.img+c2.real*c1.img;
	cout<<"Product = "<<real<<"+"<<img<<"i"<<endl;
	}
int main()
{
comp c1,c2,c3;		//objeect declarations
cout<<"Enter 1st complex number"<<endl;
```

```
c1.input();
cout<<"Enter 2nd complex number"<<endl;
c2.input();
cout<<"Addition of 2 complex numbers= "<<endl;
add(c1,c2);
cout<<"Multiplication of 2 complex numbers= "<<endl;
multi(c1,c2);
return 0;
```

```
Enter 1st complex number
Enter real and img part
3
4
Enter 2nd complex number
Enter real and img part
4
5
Addition of 2 complex numbers=
Sum = 7+9i
Multiplication of 2 complex numbers=
Product = -8+31i
```

4. Write and execute a C++ program to

a. Implement bubble sort using templates.

b. Create a C++ class that includes constructors to do the following.

i) Create an uninitialized string.

ii) Initialize an object with a string constant at the time of creation.

iii) Create an object and initialize with another object.

Also write a function to concatenate two strings.

a)

```
#include<iostream>

using namespace std;
template<class t>            //Define a template function having data type t
void bubble(t a[30],int n)     //Bubble sort function
{
	int i,j;
	t temp;              //Declare temp of data type t
	for(i=0;i<n-1;i++)
		for(j=0;j<n-i-1;j++)
	{
		if(a[j]>a[j+1])
		{
			temp=a[j];
			a[j]=a[j+1];
			a[j+1]=temp;
		}
	}//End of sorting

	cout<<"The sorted array is"<<endl;     //Display the sorted array
	for(i=0;i<n;i++)
		cout<<a[i]<<" ";
	cout<<endl;

}
main()
{
	int i,m,n,p;
	int a[30];
	char b[30];
	float c[30];

	cout<<"Enter the number of integer elements in array"<<endl;
	cin>> n;
	cout<<"Enter the integer elements"<<endl;      //Input the integer type elements
```

```
for(i=0;i<n;i++)
	cin>>a[i];
bubble(a,n);		//Bubble sort for integer data type

cout<<"Enter the number of char elements in array"<<endl;
cin>>m;
cout<<"Enter the char elements"<<endl;		//Input the char type elements
	for(i=0;i<m;i++)
		cin>>b[i];
bubble(b,m);		//Bubble sort for char data type

cout<<"Enter the number of floating type elements in array"<<endl;
cin>>p;
cout<<"Enter the floating type elements"<<endl; //Input the float type elements
	for(i=0;i<p;i++)
	cin>>c[i];
bubble(c,p);

}
```

```
Enter the number of integer elements in array

5
Enter the integer elements
-90
10
-67
5
6
The sorted array is
-90 -67 5 6 10
Enter the number of char elements in array
5
Enter the char elements
h
q
e
r
a
The sorted array is
a e h q r
Enter the number of floating type elements in array
5
Enter the floating type elements
-0.9
-.09
-0.009
-0.0009
-0.00009
The sorted array is
-0.9 -0.09 -0.009 -0.0009 -9e-05
```

b)

```
#include<iostream>
#include<string>
using namespace std;
class str1
{
string s;
public: str1(string a)
    {
    s=a;
    }
    str1(const str1 &obj)        //using const keyword
```

```
    {
    s=obj.s;
    }
    void display()
    {
    cout<<s<<endl;
    }
    friend void concat(str1 a,str1 b)    //friend function
    {
    string c;
    c=a.s+b.s;
    cout<<c<<endl;
    }
};
main()
{
str1 a("Rakshith");
str1 b=a;
a.display();
b.display();
concat(a,b);
}
```

```
Rakshith
Rakshith
RakshithRakshith
```

5. Write and execute a C++ program to create a class called 'TIME' that has - three integer data members for hours, minutes and seconds - constructor to initialize the object to zero - constructor to initialize the object to some constant value - member function to add two TIME objects - member function to display time in HH:MM:SS format. Write a main function to create two TIME objects, add them and display the result in HH:MM:SS format

```
#include<iostream>
using namespace std;
```

```
class Time
{
private:int hr,min,sec;
public: Time()                          //default constructors
        {
        hr=min=sec=0;
        }
        Time(int a,int b,int c):hr(a),min(b),sec(c)         //parameterized constructors
        {
        }
        void input()
        {
        cout<<"Enter time in hours minutes and seconds"<<endl;
        cin>>hr>>min>>sec;
        }
        void add(Time t1,Time t2)            //passing objects as parameters
        {
        sec=t1.sec+t2.sec;
        min=sec/60;
        sec%=60;
        min+=t1.min+t2.min;
        hr=min/60;
        min%=60;
        hr+=t1.hr+t2.hr;
        }
        void display()
        {
        cout<<"Resultant time= "<<hr<<"hrs"<<":"<<min<<"mins"<<":"<<sec<<"sec"<<endl;
        }
};
int main()
{
Time t1,t2(10,10,10),t3;
cout<<"Enter time 1"<<endl;
```

```
t1.input();
t3.add(t1,t2);
t3.display();
return 0;
}
```

```
Enter time 1
Enter time in hours minutes and seconds
5
3
6
Resultant time= 15hrs:13mins:16sec
```

6.Write and execute a C++ program to create a class called STUDENT with data members USN , Name and Age. Using inheritance, create the classes UGSTUDENT and PGSTUDENT having fields as Semester, Fees and Stipend. Enter the data for at least 3 students. Find the semester wise average age for all students.

```
#include<iostream>
#include<string>
using namespace std;
class student
{
    public:
        string name,id;
        int age;

    void get_PU()
    {   cout<<"Student PU info \n\n";
        cout<<"NAME ID AGE :\n";
        cin>>name;
        cin>>id;
        cin>>age;

    }
```

```
        void print_PU()
        {       cout<<"Student PU info: \n";
                cout<<"Name :"<<name<<"\n"<<"ID :"<<id<<"\n"<<"Age:"<<age<<"\n";

        }

};


class UG : public student
{
        public:
                int sem,fees,stipend;
                 static int Age[8];              // To record the sum of UG Students age Sem wise
                 static int count[8];            // To record count of UG students Sem wise


                void get_UG()
                {       get_PU();
                        cout<<"UG info \n";
                        cout<<"Enter the student SEM FEES STIPEND\n";
                        cin>>sem>>fees>>stipend;
                        cout<<"\n";

                        Age[sem-1]+=AGE();
                        count[sem-1]++;
                }

                void print_UG()
                {
                        print_PU();

                         cout<<"Student UG info: \n";
                         cout<<"SEMESTER :"<<sem<<"\n"<<"Fees:"<<fees<<"\n"<<"STIPEND"<<stipend<<"\n";

                }
```

```
        int  Sem()
        {
            return sem;

        }
        int AGE()
        {
            return age;
        }



};
int UG::Age[8]={0};        //Initializing Static members
int UG::count[8]={0};

class PG : public student
{

    public:
        int sem, fees,stipend;
         static int Age[8];    //To record the Sum of Age of Students Sem wise
        static int count[8];  //To record the Count of Students Sem wise

        void get_PG()
        {           get_PU();
                    cout<<"PG info \n";
                    cout<<"Enter SEM FEES STIPEND\n";
                    cin>>sem>>fees>>stipend;
                    cout<<"\n";

                    Age[sem-1]+=AGE();          //Adds AGE() of a student at sem-1 position
                    count[sem-1]++;             //Increments the Student count for that particualr SEM


        }
```

```
void print_PG()
{           print_PU();

            cout<<"Student PG info: \n";
            cout<<"SEM "<<sem<<"\t"<<"FEES "<<fees<<"\t"<<"STIPEND "<<stipend<<"\t \n";
}

      int  Sem()
     {
            return sem;

     }
     int AGE()
     {
            return age;
     }



};
int PG::Age[8]={0};
int PG::count[8]={0};  //Initialize Static Members

 int main()
{
      UG a[10];
      PG b[10];
      int i,m,n;

      cout<<"Enter no. of UG std \n";cin>>m;
      cout<<"Enter no.of PG std \n";cin>>n;

      for(i=0;i<m;i++)
      {     cout<<"UG :"<<i+1<<endl;
            a[i].get_UG();
```

```
    }


    if(m !=0)                   //Proceed if No. Of UG students is not zero
    {
        cout<<"UG student Details :\n";

        for(i=0;i<8;i++)            // To get Sem wise Age Avg
        {
            if(a[0].UG::count[i] !=0)      //Checking if i+1th  SEM has Students are not.If no students Count
for that SEM is 0
                cout<<"Sem : "<<i+1<<"\t"<<" Age Avg :"<<"\t"<<a[0].UG::Age[i]/a[0].UG::count[i]<<endl;

        }                   //Since static variables are class Specific,any object can be used to access it

        for(i=0;i<n;i++)
        {     cout<<"PG:"<<i+1<<endl;
            b[i].get_PG();

        }
    }

    cout<<"\n\n";

    if(n != 0)
    {
        cout<<"PG details \n";

        for(i=0;i<8;i++)
        {
            if(b[0].PG::count[i] !=0)
              cout<<"Sem : "<<i+1<<" Age Avg :"<<"\t"<<b[0].PG::Age[i]/b[0].PG::count[i]<<endl;

        }}
return 0;
}
```

```
Enter no. of UG std
2
Enter no.of PG std
2
UG :1
Student PU info

NAME ID AGE :
ramesh
1
19
UG info
Enter the student SEM FEES STIPEND
2
13000
5500

UG :2
Student PU info

NAME ID AGE :
suresh
2
18
UG info
Enter the student SEM FEES STIPEND
1
65000
6000

UG student Details :
Sem : 1  Age Avg :      18
Sem : 2  Age Avg :      19
PG:1
Student PU info

NAME ID AGE :
rakshith
1
24
PG info
```

```
2
18
UG info
Enter the student SEM FEES STIPEND
1
65000
6000

UG student Details :
Sem : 1  Age Avg :      18
Sem : 2  Age Avg :      19
PG:1
Student PU info

NAME ID AGE :
rakshith
1
24
PG info
Enter SEM FEES STIPEND
1
45000
6000

PG:2
Student PU info

NAME ID AGE :
supreeth
2
26
PG info
Enter SEM FEES STIPEND
2
45000
6000



PG details
Sem : 1 Age Avg :       24
Sem : 2 Age Avg :       26
```

7)a). Write and execute a C++ program a. That creates a binary file to hold student records. Read the data from the terminal which consists of Roll no., Name (a string of 30 or lesser no. of characters) and total marks for 3 subjects. Compute the grade and append

it to the student record and display the complete record on terminal by reading the data from binary file.

```
#include<iostream>
#include<cstdlib>
#include<fstream>      //Header providing file stream classes
using namespace std;
class student          //Declare a class named student
{
      int roll;
      int total;
      char name[50];
      char grade;
      int per;
      public:
            void setdata()        //Function to input details of student
            {
            cout<<"Enter roll number"<<endl;
            cin>>roll;
            cout<<"Enter name"<<endl;
            cin>>name;
            cout<<"Enter total marks in 3 subjects"<<endl;
            cin>>total;
            per=total/3;          //Calculate percentage to determine grade

            if(per>90)            //Assign grade depending on percentage
                  grade='S';
            else if(per>75)
                  grade='A';
            else if(per>60)
                  grade='B';
            else if(per>40)
                  grade='C';
            else
```

```
                grade='F';
            }
            void showdata()        //Function to display the student details
            {
            cout<<"Student name-"<<name<<endl;
            cout<<"Roll-"<<roll<<endl;
            cout<<"Total-"<<total<<endl;
            cout<<"Grade-"<<grade<<endl;
            cout<<endl;
            }

};

void write_record()     //Function to write record into the file
{
        //Opening a binary file named student1.bin in binary and append mode
        ofstream outfile;
        outfile.open("student1.bin", ios::binary | ios::app);

        student obj;    //Create a new object of class sudent
        obj.setdata();  //Calling the function setdata() to input the student details

        //Writing the data obtained into the binary file
        outfile.write((char*)&obj, sizeof(obj));

        //Closing the file
        outfile.close();
}

void display()         //Function to display record from file
{
        //Opening the binary file in binary and input mode
        ifstream infile;
        infile.open("student1.bin", ios::binary);
        if(!infile)            //Checking for open errors
```

```
    {
        cout<<"Error opening file."<<endl;
        exit(0);
    }
    student obj;    //Create a new objecct of student class

    //Reading the data from the file for each object
    while(infile.read((char*)&obj, sizeof(obj)))
    {
    obj.showdata();        //Displaying the student details obtained from the file
    }
    infile.close();        //Closing the file
}


int main()
{
    int n;
    cout<<"Enter the number of students"<<endl;
    cin>>n;

    //Input data depending on the number of students
    for(int i=1;i<=n;i++)
        write_record();        //Function call to write the record into file
    cout<<"List of records"<<endl;
    display();                 //Function call to display the record from file
    return 0;
}
```

```
^@A^@^@^@b^@^@^@suhas^@^@^@Ðg1^_ÿ^?^@^@^@^@^@^@^@^@^@Ìg1^_ÿ^?^@^@ö(
<83>ì^?^@^@^A^@^@^@^@^@^@^@^@F^@ ^@^@^@^A^@^@^@S^A^@^@suhas^@^@^@^PLÚ^Bþ^?^@^@^@^@^@^@^@^@^@LÚ^Bþ^?^@^@ö¸öh¾^?^@^@^A^@^@^@^@^@^@^@S^@
a^@^@^@^B^@^@^@(^A^@^@subramani^@Ú^Bþ^?^@^@^@^@^@^@^@^@^@LÚ^Bþ^?^@^@ö¸öh¾^?^@^@^A^@^@^@^@^@^@^@S^@b^@^@^@
~
~
~
~
```

7.b. Write a program having student as an abstract class and create many derived classes such as Engineering, Science, medical etc. From the student class. Create their objects and process them.

```
#include<iostream>
using namespace std;
class Student
{
public:
   virtual void f()=0;//creating a pure virtual function
   virtual void g()=0;//creating a pure virtual function
};
class Engineering:public Student
{
public:
   void f(){}
   void g()
   {

      cout<<"Welcome to engineering"<<endl;
   }
};
class Medical:public Student
{
public:
   void g(){}
   void f()
   {
      cout<<"Welcome to medical science"<<endl;
   }
};
int main()
{
   Student *s1,*s2;//creating pointers of type Student(class)
```

```
    Engineering e;//object of class Engineering
    Medical m;//object of class Medical
    s1=&e;
    s2=&m;
    s1->g();
    s2->f();
    return 0;
}
```

```
Welcome to engineering
Welcome to medical science
```

8.Create a C++ class RATIONAL which represents a numerical value by two double values NUMERATOR & DENOMINATOR. Include the following public member Functions:

a. Constructor with no arguments (default).

b. Constructor with two arguments.

c. void reduce( ) that reduces the rational number by eliminating the highest common factor between the numerator and denominator.

d. Overload >> operator to enable input through cin.

e. Overload << operator to enable output through cout.

Write and execute a main ( ) to test all the functions in the c

```
#include<iostream>
using namespace std;

class rational
{
      int nem,deno;
```

```
    public:
        rational()
        {
            nem=0;
            deno=1;
        }

        rational(int a,int b)
        {   nem=a;
            deno=b;
        }

        void reduce()
        {   int gcd;
            int rem,i;
            int n1=nem,n2=deno;

                                //GCD

             for(i=1; i <= n1 && i <= n2; ++i)
            {
                                // Checks if i is factor of both integers
            if(n1%i==0 && n2%i==0)
                gcd = i;
            }
            nem/=gcd;               //Dividing Numerator and Denominator by GCD
            deno/=gcd;
        }


        friend ostream  &operator << (ostream &output,rational &p);
        friend istream &operator >> (istream &input,rational &p);

};
```

```
ostream &operator << (ostream &output,rational &p)          //Operator Overloading
{
    output<<p.nem<<"/"<<p.deno<<endl;
    return output;
}
istream &operator >> (istream &input,rational &p)
{
    cout<<"Enter enter a rational num in new & deno form \n";
    input>>p.nem>>p.deno;
    return input;
}


int main()
{
    rational r(8,4);
    cin>>r;
    r.reduce();
    cout<<r;


    return 0;
}
```

```
Enter enter a rational num in new & deno form
2
5
2/5
```

9.. a. Write and execute a C++ program to implement the following inheritance. A base class called person. A teacher class which inherits basic information from Person. One more class called Student also inherits from Person class. An additional class called marks becomes a derived class of Student

Assume suitable data members and member functions for all the classes. Display the number of publications for a teacher and read three test marks in student class and display the percentage marks for a student in marks class.

```
#include<iostream>

#include<string>
using namespace std;
class Person //creating the base class person
{
private:
string name; //declaring his name and age
int age;
public:
void get()

{

cout<<"enter the age"<<endl;

cin>>age;

cout<<"enter his name"<<endl;

cin>>name;

}

void display()

{

cout<<"name and age is "<<name<<age;
} //display name and age
};
class teacher:public Person     // publicly inheriting teacher from person
```

```
{
private:
int n;
string des;
public:
void get1()
{
get(); //calling the base class input function
cout<<"Enter designatiom"<<endl;
cin>>des;
cout<<"Enter the no of publications"<<endl;
cin>>n;
}
void disp()
{
display(); // calling display function of base class
cout<<"Designation-"<<des<<"\nNo of publications-"<<n<<endl;
}
};
class Student:public Person    // publicly inheriting student from person
{
private:
int roll;
public:
void get2()
{
get(); // calling input function of base class




cout<<"Enter the roll number"<<endl;
cin>>roll;
}
```

```
void disp1()
{
disp();
cout<<"Roll number-"<<roll<<endl;
}
};
class Marks:public Student    // publicly inherit marks from student class
{
int m1,m2,m3;
float perc;
public:
void  get3()
{
get2();// call input function of student class
cout<<"Enter the marks in 3 subjects"<<endl;
cin>>m1>>m2>>m3;
perc=(m1+m2+m3)/3;
}
void disp2()
{
disp1();// call display function of student class
cout<<"Percentage="<<perc<<endl;
}
};
int main()
{
marks s;// create object of marks class
teacher t;// create object of teacher class
cout<<"Enter the student details"<<endl;
s.get3();
cout<<"Enter the teacher details"<<endl;
t.get1();
s.calc();
s.disp2();
```

```
return 0;
}
```

```
enter the details of :
 1.Teacher
2.Student
enter your choice
1
enter name,age,branch,college
ramesh
35
mech
svit
enter number of publications
10
name: ramesh
age: 35
branch: mech
college:svit
number of publications 10
```

9b. Write and execute a C++ program to illustrate the order of execution of constructor and destructors using multiple inheritance and multilevel inheritance.

```
#include<iostream>
using namespace std;
class class1 //create class called class1
{
public:class1() // call constructor of that class
{
cout<<"constuctor1"<<endl;
}
~class1() // call destuctor
{
cout<<"destuctor1"<<endl;
}
};
class class2:public class1 // inherit from class1
{
public:class2() // constructor of class2
```

```
{
cout<<"constuctor2"<<endl;
}
~class2() // destructor of class2
{
cout<<"destuctor2"<<endl;
}
};
int main()
{
class2 c1; // create object of derived class
return 0;
}
```

```
constuctor1
constuctor2
destuctor2
destuctor1
```

10) a)Demonstrate the use of multiple try and catch blocks. Also demonstrate the re-throwing of the caught exceptions.

```
#include<iostream>
#include<stdexcept>
using namespace std;

int main()
{
	try{
	try
	{
		throw " A Char Exception ";		//Throws a String
	}
	catch(const char *a)
	{
```

```
            cout<<"Char type in inner block"<<endl;
            cout<<"Rethrowing the exception "<<endl;

            throw;
        }
        }
        catch(const char *a)

{

cout<<"Char type in outer block"<<a<<endl;

}
        catch(...)

{

cout<<"Unexpected exception in the outerblock"<<endl;
       }
return 0;
}
```

```
Char type in inner block
Rethrowing the exception
Char type in outer block A Char Exception
```

10) b)Consider a Bookshop which sells both book and video_tapes, Create a class known as MEDIA that stores the Title and Price of publication then create two derived class, one for storing the number of pages in a book and another for storing the play time of the tape using the concept of pure virtual function and passing parameters to base class.

```
#include<iostream>
using namespace std;

class MEDIA
{
```

```
        public:
                string title,price;

                MEDIA(string a,string b)
                {       title=a;
                        price=b;
                }

                 virtual void content(string)=0;        //Pure virtual fns
                 virtual void getinfo()=0;

                ~MEDIA(){ }


};

class Book : public MEDIA
{       private : string no_pgs;

        public:

        Book(string a,string b):MEDIA(a,b){ }           //To invoke base class constructor


        void content(string s){                         //Defining the Virtual fn
                no_pgs=s;
        }

        void getinfo()
        {
           cout<<"Title :"<<title<<" "<<"Price :"<<price<<endl;
           cout<<"No.of Pgs :"<<no_pgs<<endl;
        }
```

```
};


class Tape : public MEDIA
{       public:

        string play_time;

        Tape(string a,string b):MEDIA(a,b){ }          //To invoke base class constructor


        void content(string s)
        {
                play_time=s;
        }

        void getinfo()
        {       cout<<"Title :"<<title<<" "<<"Price :"<<price<<endl;

                cout<<"Playtime: "<<play_time<<endl;
        }

};

void access( MEDIA &m)                          //Illustrates Run time Polymorphism
{
  m.getinfo();
}

int main()
{
   Book B("HARRY POTTER","Rs.750");
   Tape T("Yesterday by Beatles","Rs.50");

        T.content("12:20");
        B.content("250");
```

```
    access(T);
     access(B);

     return 0;
}
```

```
Title :Yesterday by Beatles Price :Rs.50
Playtime: 12:20
Title :HARRY POTTER Price :Rs.750
No.of Pgs :250
```

11. a. Write and execute a C++ program using STL to

i. Insert an element into the vector

ii. Delete the last element from the vector

iii. Display the Size of vector

iv. Display the elements in vector

v. Clear the vector

```
#include<iostream>

 #include<vector>

using namespace std;

main()

{

vector<int>vector1;

 int ch,m,n;

 do
```

```
 {

 cout<<"\nEnter \n1>To insert \n 2>Delete the last element \n3>Size of vector \n 4>Display elements in
vector \n5>Clear the vector\n6>Exit\n";

 cin>>ch; //user is asked to choose an option from the above

switch(ch)

 {

 case 1:cout<<"\nEnter the element to be inserted\n";

          cin>>m; vector1.push_back(m); //element is inserted at the end in vector1

         break;

case 2:vector1.pop_back();

         cout<<"\nElement deleted\n";

       break; //last element is deleted from vector1

 case 3:cout<<"\nSize of vector is:\n";

          cout<<vector1.size(); //displays number of elements in vector1

         break;

case 4:cout<<"\nElements of vector are:\n";

         if(vector1.size()==0)

         cout<<"Vextor is empty";

          else {

         for(n =0;n <vector1.size();n++)
```

```
        cout<<vector1[n]<<"\t";

       } //if there are elements in the vector,itdisplays the elements from beggining to end or else itdisplays
the vector

        break;

 case 5:vector1.clear();

          cout<<"\nThe vector has been cleared\n"; break; //vector1 is cleared case 6:break;

  }

}while(ch!=6);

}
```

```
Enter
1>To insert
 2>Delete the last element
3>Size of vector
 4>Display elements in vector
5>Clear the vector
6>Exit
1

Enter the element to be inserted
2

Enter
1>To insert
 2>Delete the last element
3>Size of vector
 4>Display elements in vector
5>Clear the vector
6>Exit
1

Enter the element to be inserted
3

Enter
1>To insert
 2>Delete the last element
3>Size of vector
 4>Display elements in vector
5>Clear the vector
6>Exit
4

Elements of vector are:
1       2       3
Enter
1>To insert
 2>Delete the last element
3>Size of vector
 4>Display elements in vector
5>Clear the vector
6>Exit
```

b . Create a class called TEST with a data member Code and Static data member Count and static member function showcount. Write and execute a C++ program to display the code number for each of the object.

```
#include<iostream>
```

```
using namespace std;

class TEST

 {

int code;

 static int count;

 public: static void showcount(int c,TEST ob)

             {

             cout<<count<<"";

           ob.code =c;

          cout<<ob.code<<endl; count++;

             }

 };

int TEST :: count = 1;

main()

{

TEST ob,ob1;

 int c;

cout<<"Enter the code for object1\n";

 cin>>c;

TEST ::showcount(c,ob);
```

```
 TEST ::showcount(20,ob1);

}
```

```
Enter the code for object1
123
1123
220
```

12.a Write and execute a C++ Program to

a. Find the Number of Lines in a Text File.

```
#include<iostream>
#include<fstream>
#include<cstdlib>
using namespace std;

int main()
{
	ifstream   fin("Newfile.txt",ios::in); //open file in Read mode only
	string s;
	int count=0;

	if(!fin)
	{
		cout<<"Error file doesnt exist \n";
		exit(0);
	}


	while(getline(fin,s))		//getline also reads whitespace up until \n is encountered
	{
		count++;
	}
```

```
    cout<<"No. of lines :"<<count<<endl;


}
```

```
sdasdas
asdasd
asd
asd
a
sda
sd
asd
asd
as
dsa
d
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"Newfile.txt" 12L, 49C
```

```
akhilesh@akhilesh-HP-Pavilion-Notebook:~/Programs/cplusplus/final_lab_programs$ vi Newfile.txt
akhilesh@akhilesh-HP-Pavilion-Notebook:~/Programs/cplusplus/final_lab_programs$ ./a.out
No. of lines :12
akhilesh@akhilesh-HP-Pavilion-Notebook:~/Programs/cplusplus/final_lab_programs$
```

12.b Write and execute a C++ Program to

i. Append the Content of a File at the end of another file.

```
#include<iostream>
#include<fstream>
#include<cstdlib>
using namespace std;

int main()
{
	ifstream fin("Newfile.txt",ios::in);	//File to be read from
	ofstream fout("Secondfile.txt",ios::out | ios::app); //File to be appended to
	string s;

	if(!fin)
	{
		cout<<"File doesnt exist \n";
		exit(0);
	}

	while(getline(fin,s))	//Reads a line from file pointed by fin into s
		fout<<s;	//Write's the string into file pointed by fout

}
```

13.a Write and execute a C++ program using STL to

i. Insert an Element into the Map

ii. Delete the Element from the Map

iii. Find Element at a key in a Map

iv. Display value of an Element at a specific key

v. Size of the Map

vi. Display by using iterator

```
#include<iostream>
#include<map>
using namespace std;
int main()
{
	map<char,string> m;
	map<char,string>::iterator it;
	int n,i;
	char a;
	string s;
	cout<<"Enter no. of map elements \n";
	cin>>n;

	for( i=1;i<=n;i++)	//To insert elements
	{
		cout<<" KEY-VALUE :";
		cin>>a;
		cin>>s;
		m.insert(pair<char,string>(a,s));	//OR m[a]=s;
	}


	//To Delete the element
	cout<<"Enter the Value\n";
	cin>>s;
	it=m.begin();
	for( ;it!=m.end();it++)
	{
		if(it->second==s)
		{	m.erase(it);
			break;
```

```
        }
    }

    if(it == m.end())
        cout<<"Element not found \n";
    else
        cout<<"Item found \n";

    //Finding element using key
    cout<<"Enter the key \n";
    cin>>a;
    it=m.find(a);

    if(it != m.end())
        cout<<it->second<<endl<<endl;
    else
        cout<<"Element not found \n\n";


    cout<<m.size()<<endl<<endl;


    //To display contents of map.
    for(it=m.begin();it!=m.end();it++)
    {
        cout<<"Key="<<it->first<<" ITEM="<<it->second<<endl;
    }
}
```

```
Enter no. of map elements
6
 KEY-VALUE :1
15
 KEY-VALUE :2
16
 KEY-VALUE :3
17
 KEY-VALUE :4
18
 KEY-VALUE :5
19
 KEY-VALUE :6
20
Enter the Value
1
Element not found
Enter the key
1
15

6

Key=1 ITEM=15
Key=2 ITEM=16
Key=3 ITEM=17
Key=4 ITEM=18
Key=5 ITEM=19
Key=6 ITEM=20
```

13.b Write and execute a C++ program to read a paragraph from a text file and find the count of the vowels individually for each of the a,e,i,o and u for both the cases (Uppercase & Lowercase). Display theCount of the vowels individually and position at which they are found in another file.

```
#include <iostream>
#include<fstream>
using namespace std;
int main()
{
   const int SIZE = 100;
   char stringObj[SIZE];
```

```
 int i= 0;
fstream file("new.txt",ios::in);
 int valueAt = 1;
 int check = 0;
 int count=0;
 file.getline(stringObj,3000);
 while(stringObj[i] !='\0')
 {
    switch(stringObj[i])
    {
      case 'A':
      case 'a':
      case 'E':
      case 'e':
      case 'I':
      case 'i':
      case 'O':
      case 'o':
      case 'U':
      case 'u':
         check =1;
         cout<<endl;
      if(check == 1)
         {
       cout<<"Vowel "<<stringObj[i]<<" found at index number "<<valueAt<<endl; //printing the index
of the vowels
       count++;                              //counts the number of  vowels in the string
      }
      check = 0;
    }
   valueAt++;
   i++;
 }
          cout<<stringObj;
```

```
   cout<<"\nTotal number of vowels ="<<count<<endl;
file.close();
return 0;
}
```

```
slkjlnlaksaeiouansldknlladgvcrqwe
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"new.txt" 2L, 35C                                                    1,1          All
```

```
Enter a String to find vowel:   Vowel a found at index number 8
Vowel a found at index number 11
Vowel e found at index number 12
Vowel i found at index number 13
Vowel o found at index number 14
Vowel u found at index number 15
Vowel a found at index number 16
Vowel a found at index number 25
Vowel e found at index number 33

Total number of vowels =9
```

14. Write a C++ program to perform operations on list using STL to

a. Insert an Element at the Front & at the End

b. Delete the Element at the Front & at the End

c. Size of the List

d. Remove Elements with Specific Values & duplicate values

e. Reverse the order of elements

f. Merge & display Sorted Lists

```
#include <iostream>

 #include <list>

 using namespace std;

int main() //Main function

 {

 int choice, a,b;

 list<int> A; list<int> B; //Declaration of the lists

//Inserting values into list B

 B.push_back(10);

 B.push_back(20);

 B.push_back(30);

// Iterator from front to end of the list

 list<int>::iterator i=A.begin();

do {

 //Menu

cout<<"\nOperations on list"<<endl;
```

```
 cout<<"1-Insert an element at front and end \n";

cout<<"2-Delete an element from front and end \n";

 cout<<"3-Size of list\n";

 cout<<"4-Remove element withspecific values and duplicate values \n ";

 cout<<"5-Reverse the list\n ";

 cout<<"6-Merge and sort the list \n ";

cout<<"7-Display the list\n 8-Exit\n";

cout<<"Enter your choice: ";

 cin>>choice;

cout<<endl;

 switch(choice)

 {

// To insert values at front and end of list

 case 1: cout<<"Element to be inserted at front: ";

            cin>>a;

             cout<<"Element to be inserted at end :";

             cin>>b;

             A.push_front(a);

             A.push_back(b);

            break;
```

```
 case 2: A.pop_front(); //To delete values from front and end

            A.pop_back();

            cout<<"Elements deleted from front and back"<<endl;

             break;

 //To find sizeof the list

 case 3: cout<<"Size of the listis: "<<A.size()<<endl;

            break;

 //To delete a specific value and repeated values from the list

case 4: cout<<"Enter the value to be deleted: ";

            cin>>a;

            A.remove(a);

             A.unique();

            cout<<"Value deleted"<<endl;

            break;

//To reverse the list

case 5: cout<<"The reversed listis:"<<endl;

            A.reverse();

            for (i =A.begin(); i!= A.end(); i++)

            cout<<endl;

            break;
```

```
 //To merge two listsand sort it

case 6: //Display list A cout<<"ListA:";

                 for (i =A.begin(); i!= A.end(); i++)

                cout<<endl;

                //Display list B cout<<"ListB:";

                 for (i =B.begin(); i!= B.end(); i++)

                 cout<<*i<<"\t";

                cout<<*i<<"\t";

                cout<<*i<<"\t";

                cout<<endl<<endl;

                A.merge(B);

                 A.sort();

                //Display merged and sorted list

                cout<<"Merged and sorted list:"<<endl;

                for (i =A.begin(); i!= A.end(); i++)

                 cout<<*i<<"\t"; cout<<endl;

                break;

                //To display the list

case 7: cout<<"The listis:\n";

                for (i =A.begin(); i!= A.end(); i++)
```

```
            cout<<*i<<"\t";

             cout<<endl;

             break;

//Case exit

case 8: break;

//Invalid choice

default: cout<<"Invalid choice"<<endl;

 }

}//End of switch

 while (choice !=8); // End of do while

 } //End of main
```

```
Enter your choice: 1

Element to be inserted at front: 1
Element to be inserted at end :2

Operations on list
1-Insert an element at front and end
2-Delete an element from front and end
3-Size of list
4-Remove element withspecific values and duplicate values
 5-Reverse the list
 6-Merge and sort the list
 7-Display the list
 8-Exit
Enter your choice: 1

Element to be inserted at front: 3
Element to be inserted at end :4

Operations on list
1-Insert an element at front and end
2-Delete an element from front and end
3-Size of list
4-Remove element withspecific values and duplicate values
 5-Reverse the list
 6-Merge and sort the list
 7-Display the list
 8-Exit
Enter your choice: 7

The listis:
3       1       2       4

Operations on list
1-Insert an element at front and end
2-Delete an element from front and end
3-Size of list
4-Remove element withspecific values and duplicate values
 5-Reverse the list
 6-Merge and sort the list
 7-Display the list
 8-Exit
Enter your choice:
```

```
Enter your choice: 4

Enter the value to be deleted: 2
Value deleted

Operations on list
1-Insert an element at front and end
2-Delete an element from front and end
3-Size of list
4-Remove element withspecific values and duplicate values
 5-Reverse the list
 6-Merge and sort the list
 7-Display the list
 8-Exit
Enter your choice: 7

The listis:
3       1       4

Operations on list
1-Insert an element at front and end
2-Delete an element from front and end
3-Size of list
4-Remove element withspecific values and duplicate values
 5-Reverse the list
 6-Merge and sort the list
 7-Display the list
 8-Exit
Enter your choice: 7

The listis:
3       1       4

Operations on list
1-Insert an element at front and end
2-Delete an element from front and end
3-Size of list
4-Remove element withspecific values and duplicate values
 5-Reverse the list
 6-Merge and sort the list
 7-Display the list
 8-Exit
Enter your choice:
```

15) Write and execute a C++ Program to implement stack with necessary exception handling

```
#include<iostream>
#include<cstdlib>      //Header for using exit function
using namespace std;
class stack            //Create a class named stack
{
      private:
            int *s;        //Declare stack as pointer
            int max;       //Declare max and top
            int top;
      public:
            class full{};
            class empty{};
            stack(int);
            void push(int);
            int pop();
            void display();
};

stack::stack(int m)            //Constructor for creating a new stack
{
      s=new int[m];          //Declare s as an array of max size m
      top=-1;                //Intialize top to -1
      max=m;                 //Assign max to m
}

//Function to push an element into the stack
void stack::push(int item)
{
      if(top<max-1)          //Insert element at top of stack if stack is not full
            s[++top]=item;
      else
            throw full();   //If stack is full throw the exception full()
}
```

```
//Function to pop an element from stack
int stack::pop()
{
	if(top>=0)		//If stack is not empty then remove and return element at top of stack
		return s[top--];
	else
		throw empty();  //If stack is empty throw the exception  empty()
}

//Function to display contents of the stack
void stack::display()
{
	int i;
	if(top>=0)			//Dispalying contents of the stack
	{
		for(i=top;i>=0;i--)
			cout<<endl<<s[i];
		cout<<endl;
	}

	else				//If stack is empty throw the exception empty()
		throw empty();
}


int main()
{
	int item,size;
	int ch;
	int item1;
	cout<<"Enter the size of the stack"<<endl;
	cin>>size;
	stack s1(size);		//Create an object of class stack with parameter as stack size
```

```
//Implementation of stack operations
cout<<"MENU"<<endl<<"1.Push 2.Pop 3.Display 4.Eit"<<endl;
cout<<"Enter your choice"<<endl;       //Obtain the choice from user
cin>>ch;
while(1)
{
switch(ch)                    //ch is the choice entered by user
{
        case 1:
                cout<<"Enter the item to push"<<endl;
                cin>>item;
                try            //Put the push function under try block to check for exceptions
                {
                s1.push(item);
                }
                catch(stack::full)      //If exception is present execute catch block statements
                {
                cout<<"Stack overflow"<<endl;
                }
                break;
        case 2:
                try                    //Put the pop function under try block to check for exceptions
                {
                item1=s1.pop();
                cout<<"The popped element is "<<item1<<endl;
                }
                catch(stack::empty)            //If exception is present execute catch block statements
                {
                cout<<"Stack is empty"<<endl;
                }
                break;
        case 3:
                cout<<"The stack is";
                try                    //Put the display function under try block to check for exceptions
```

```
                {
                s1.display();
                }
                catch(stack::empty)     //If exception is present execute catch block statements
                {
                cout<<" empty"<<endl;
                }
                break;
            case 4:
                exit(0);            //Exit the program
        }//End of switch

cout<<"Enter your choice"<<endl;        //Obtain the choice from user
            cin>>ch;
        }//End of while
        return 0;

}
```

```
Enter the size of the stack
10
MENU
1.Push 2.Pop 3.Display 4.Eit
Enter your choice
1
Enter the item to push
1
Enter your choice
1
Enter the item to push
5
Enter your choice
2
The popped element is 5
Enter your choice
1
Enter the item to push
7
Enter your choice
2
The popped element is 7
Enter your choice
3
The stack is
1
Enter your choice
1
Enter the item to push
2
Enter your choice
3
The stack is
2
1
Enter your choice
4
```