**Q1/\*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

{float x; float y;

public: complex() {

x=0; y=0; }

complex operator+(complex); complex operator\*(complex);

friend istream &operator >>(istream &input,complex &t) {

cout<<"Enter the real part"; input>>t.x;

cout<<"Enter the imaginary part";

input>>t.y; }

friend ostream &operator <<(ostream &output,complex &t) {

output<<t.x<<"+"<<t.y<<"i\n"; } };

complex complex::operator+(complex c) {

complex temp; temp.x=x+c.x; temp.y=y+c.y; return(temp); }

complex complex::operator\*(complex c) {

complex temp2; temp2.x=(x\*c.x)-(y\*c.y);

temp2.y=(y\*c.x)+(x\*c.y);

return (temp2); }

int main() { complex c1,c2,c3,c4;

cout<<"Default constructor value=\n";

cout<<c1; cout<<"\nEnter the 1st number\n";

cin>>c1; cout<<"\nEnter the 2nd number\n";

cin>>c2; c3=c1+c2; c4=c1\*c2;

cout<<"\nThe first number is ";

cout<<c1; cout<<"\nThe second number is ";

cout<<c2; cout<<"\nThe addition is ";

cout<<c3; cout<<"\nThe multiplication is ";

cout<<c4; return 0; }

**/\*output**

student@student-OptiPlex-3010:~$ ./a.out

Default constructor value=0+0i

Enter the 1st number Enter the real part2

Enter the imaginary part4 Enter the 2nd number

Enter the real part4 Enter the imaginary part8

The first number is 2+4i The second number is 4+8i

The addition is 6+12i The multiplication is -24+32i

student@student-OptiPlex-3010:~$

\*/

**Q2/\*Implement a class CppArray which is identical to a one-dimensional C++ array (i.e.,**

**the index set is a set of consecutive integers starting at 0) except for the following :**

**1. It performs range checking.**

**2. It allows one to be assigned to another array through the use of the**

**assignment operator (e.g. cp1= cp2)**

**3. It supports a function that returns the size of the array.**

**4. It allows the reading or printing of array through the use of cout and cin.**

**\*/**

#include<iostream>

using namespace std;

class CppArray

{

private:

int a[10],i,j,n,b[10],temp;

public:

void get(); void print();

void sort(); void range();

void exchange(); int size();};

void CppArray::get() {

cout<<"\n Enter the limit of array ";

cin>>n;

cout<<"\n Enter the array elements ";

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

cout<<"\n a["<<i<<"]=";

cin>>a[i]; } } void CppArray::print() {

cout<<"\n The entered array is ";

for(i=0;i<n;i++) { cout<<"\n a["<<i<<"]="<<a[i];} }

void CppArray::sort() { for(j=0;j<n;j++) {

for(i=0;i<n-1;i++) { if(a[i]>a[i+1]) {

temp=a[i+1]; a[i+1]=a[i];

a[i]=temp; } } }

cout<<"\n The sorted array is ";

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

cout<<"\n a["<<i<<"]="<<a[i]; } }

void CppArray::range(){

cout<<"\n The range of array is from "<<a[0]<<" to "<<a[n-1];

} void CppArray::exchange()

{ for(i=0;i<n;i++) { b[i]=a[i]; }

cout<<"\n The exchanged array is ";

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

cout<<"\n b["<<i<<"]="<<b[i]; }}

int CppArray::size()

{return n; } int main()

{ int a;

CppArray obj; obj.get();

obj.print(); obj.range();

obj.exchange(); obj.sort();

a=obj.size();

cout<<"\n The size of array is "<<a;

return 0;}

**/\*Output:**

[student@localhost ~]$ g++ Array.cpp

[student@localhost ~]$ ./a.out

Enter the limit of array 5

Enter the array elements

a[0]=2 a[1]=6 a[2]=10

a[3]=1 a[4]=21

The entered array is

a[0]=2 a[1]=6

a[2]=10 a[3]=1 a[4]=21

The range of array is from 2 to 21

The exchanged array is

b[0]=2 b[1]=6 b[2]=10

b[3]=1 b[4]=21

The sorted array is

a[0]=1 a[1]=2

a[2]=6 a[3]=10

a[4]=21

The size of array is 5[student@localhost ~]$

\*/

**Q3**

**\*Write a C++ program create a calculator for an arithmetic operator (+, -, \*, /). The program should take two operands from user and performs the operation on those two operands depending upon the operator entered by user. Use a switch statement to select the operation. Finally, display the result. Some sample interaction with the program might look like this:**

**Enter first number, operator, second number: 10 / 3**

**Answer = 3.333333**

**Do another (y/n)? y**

**Enter first number, operator, second number: 12 + 100**

**Answer = 112**

**Do another (y/n)? n**

**\*/**

#include<iostream>

using namespace std;

class Calculator

{ private:

float num1,num2,result; char op;

public: void get(); void calculate();

}; void Calculator::get()

{ cout<<"\nEnter first number, operator, second number:\n";

cin>>num1; cin>>op;

cin>>num2; }

void Calculator::calculate()

{ switch(op) { case '+':

result=num1+num2;

cout<<" Answer = "<<result;

break;

case '-': result=num1-num2;

cout<<" Answer = "<<result;

break;

case '\*': result=num1\*num2;

cout<<" Answer = "<<result;

break;

case '/': if(num2==0)

cout<<"\n Error. Not valid.";

result=num1/num2;

cout<<" Answer = "<<result;

break; } }

int main() {

char ag; Calculator obj;

x:obj.get(); obj.calculate();

cout<<"\n Do another (y/n)? ";

cin>>ag; if(ag=='y'||ag=='Y')

goto x; return 0;

}

**/\*OUTPUT:**

**student@student-OptiPlex-3010:~$ g++ groupa5.cpp**

**student@student-OptiPlex-3010:~$ ./a.out**

**Enter first number, operator, second number:**

**10/3 Answer = 3.33333**

**Do another (y/n)? y**

**Enter first number, operator, second number:**

**12+100**

**Answer = 112**

**Do another (y/n)? n\*/**

**Q4/\*Develop an object oriented program in C++ to create a database of student information system containing the following information: Name, Roll number, Class, division, Date of Birth, Blood group, Contact address, telephone number, driving license no. etc Construct the database with suitable member functions for initializing and destroying the data viz constructor, default constructor, Copy constructor, destructor, static member functions, friend class, this pointer, inline code and dynamic memory allocation operators-new and delete.\*/**

#include<iostream>

#include<string.h>

using namespace std;

class person\_additional\_info

{char address[20],license[20],insurance[20];

long int contact; public:

person\_additional\_info() //Default constructor

{strcpy(address,"XYZ"); strcpy(license,"XY-0000000000");

strcpy(insurance,"XY00000000X");

contact=000000000;}

~person\_additional\_info() //Destructor

{cout<<"I am in Destructor"; }

friend class person; // Declaration Friend class

};//Definition of friend class

class person{

char name[20], dob[10], blood[10];

float ht,wt; static int count; // Static variable

person\_additional\_info \*pai;

public: person() //Default constructor {

strcpy(name,"XYZ");

strcpy(dob,"dd/mm/yy");

strcpy(blood,"A +");

ht=0; wt=0;

pai=new person\_additional\_info;

}person(person\*p1) //Copy constructor

{ strcpy(name,p1->name);

strcpy(dob,p1->dob);

strcpy(blood,p1->blood);

ht=p1->ht; wt=p1->wt;

pai=new person\_additional\_info;

strcpy(pai->address,p1->pai->address);

strcpy(pai->license,p1->pai->license);

strcpy(pai->insurance,p1->pai->insurance);

pai->contact=p1->pai->contact}

static void showcount() //Static member function

{cout<<"\nNo of records count="<<count<<"\n";

}~person() //Destructor {

cout<<"\nI am in Destructor\n" }

void getdata(char name[20]);

inline void display(); // Inline function declaration

};void person::getdata(char name[20]){

strcpy(this->name,name); //this pointer

cout<<"\n Enter date of birth";

cin>>dob; cout<<"\n Enter blood group";

cin>>blood;

cout<<"\n Enter height";

cin>>ht; cout<<"\n Enter weight";

cin>>wt; cout<<"\n Enter address";

cin>>pai->address;

cout<<"\n Enter Licence number";

cin>>pai->license;

cout<<"\n Enter Insurance policy number";

cin>>pai->insurance;

cout<<"\n Enter Contact number";

cin>>pai->contact; count++;

}//inline function definition

void person::display(){

cout<<"\t"<<name; cout<<"\t"<<dob;

cout<<"\t"<<blood; cout<<"\t"<<ht;

cout<<"\t"<<wt; cout<<"\t"<<pai->address;

cout<<"\t"<<pai->license;

cout<<"\t"<<pai->insurance; cout<<"\t"<<pai->contact;

}int person::count; //Static variable definition

int main(){person \*p1,\*p2;

int ch;p1=new person; //call default constructor & dynamic memory allocation

p2=new person(p1); //call copy constructor

cout<<"\tName"; cout<<"\tDob";

cout<<"\t Blood"; cout<<"\tHt";

cout<<"\tWt"; cout<<"\tAddress";

cout<<"\tLicense"; cout<<"\tInsurance";

cout<<"\tContact"; cout<<endl;

cout<<"Default Constructor Value \n";

p1->display(); cout<<"\n";

cout<<"Copy Constructor Value \n";

p2->display(); int n;

cout<<"\nEnter how many records you want??";cin>>n;

person p3[n]; //array of object

char name[20]; int x=0;

do{

cout<<"\nWelcome to Personal database system";

cout<<"\n1.Enter the record";

cout<<"\n2.Display the record";

cout<<"\n3.Exit";

cin>>ch; switch(ch){

case 1: {

cout<<"\nEnter the Name ";

cin>>name; p3[x].getdata(name);

person::showcount(); // calls static function

x++; break; }

case 2:{cout<<"\tName";

cout<<"\tDob"; cout<<"\t Blood";

cout<<"\tHt"; cout<<"\tWt";

cout<<"\tAddress"; cout<<"\tLicense";

cout<<"\tInsurance"; cout<<"\tContact";

for(int i=0;i<n;i++){cout<<"\n";

p3[i].display(); //calls inline function}

break;} }

}while(ch!=3);

delete p1; //dynamic memory de-allocation

delete p2; return 0;}

**Q5Imagine 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>

#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;

**Q6 Write a function in C++ to count and display the number of lines not starting with alphabet 'A' present in a text file "STORY.TXT".**

**October 19, 2018**

**Write a function in C++ to count and display the number of lines not starting with**

**alphabet 'A' present in a text file "STORY.TXT".**

**Example:**

**If the file "STORY.TXT" contains the following lines,**

**The roses are red.**

**A girl is playing there.**

**There is a playground.**

**An aeroplane is in the sky.**

**Numbers are not allowed in the password.**

**The function should display the output as 3.**

#include<iostream>

#include<fstream>

using namespace std;

int main()

{

ifstream fin;

fin.open("Story.txt");

char str[100]; int count=0;

int count1=0;

while(!fin.eof())

{

fin.getline(str,100);

if(str[0]!='A'){

count++;}

else if(str[0]=='A'&&str[1]==' '){

count1++; }}

cout<<"The number of lines not starting with 'A' are:"<<count<<"\n";

cout<<"The number of lines starting with 'A' are:"<<count1<<"\n";

fin.close(); return 0;}

exit(0);}default:

{cout<<"\n Invalid";}

}}while(choice!=5);

return 0;}

**Q7 Implement C++ program to write a class template to represent a generic**

**vector. Include following member functions:**

**To create the vector.**

**To modify the value of a given element**

**To multiply by a scalar value**

**To display the vector in the form (10,20,30,...)**

#include<iostream> //header file

#include<vector>

using namespace std; //declared scope of program

void display (vector<int>&v) //function declaration{

for(int i=0;i<v.size();i++){cout<<" "<<v[i];}

cout<<" \n";}

int main()

{vector<int> v;

cout<<"\n Initial Size: "<<v.size();

int x;

cout<<"\n Enter The 5 Element: ";

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

cin>>x;

v.push\_back(x);}

cout<<"\n Size After Insertion: "<<v.size();

cout<<"\n Vector Element: ";

display(v);

cout<<"\nVector Element After insertion of 3 at End of Vector: ";

v.push\_back(3); display(v);

vector<int>::iterator itr=v.begin();

itr=itr+3; v.insert(itr,9);

cout<<"\n Content After Insertion 9 at position 3rd: ";

display(v);

v.erase(v.begin()+3,v.begin()+5);

cout<<"\n After Erasing(3rd-4th position): ";

display(v);}

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

gescoe@gescoe-OptiPlex-3010:~/Desktop/se$ g++ fifth.cpp

gescoe@gescoe-OptiPlex-3010:~/Desktop/se$ ./a.out

Initial Size: 0

Enter The 5 Element: 6 9 4 2 1

Size After Insertion: 5

Vector Element: 6 9 4 2 1

Vector Element After insertion of 3 at End of Vector: 6 9 4 2 1 3

Content After Insertion 9 at position 3rd: 6 9 4 9 2 1 3

After Erasing(3rd-4th position): 6 9 4 1 3