Lab Work C++ Solution

2018

AL-AKHIR NAYAN

ID- 142014029

14-Aug-18

***Lab Work 1***

1.

#include <iostream>

using namespace std;

class Factorial

{

public:

void calculation( int n)

{

int factorial =1;

for(int i = 1; i <=n; ++i)

{

factorial \*= i;

}

cout << "Factorial of " << n << " = " << factorial;

}

};

int main()

{

int n;

cout << "Enter a positive integer: ";

cin >> n;

Factorial obj;

obj.calculation(n);

return 0;

}

2.

#include <iostream>

using namespace std;

class PrimeNumber

{

public:

void Prime\_Calculation(int num1, int num2)

{

cout << "Prime numbers between " << num1 << " and " << num2 << " are: ";

while (num1 < num2)

{

int flag = 0;

for( int i = 2; i <= num1/2; ++i)

{

if(num1 % i == 0)

{

flag = 1;

break;

}

}

if (flag == 0)

{

cout << num1 << " ";

}

++num1;

}

}

};

int main()

{

int number1, number2;

cout << "Enter two numbers(Integer): ";

cin >> number1 >> number2;

PrimeNumber obj;

obj.Prime\_Calculation(number1,number2);

return 0;

}

3.

#include <iostream>

using namespace std;

class Calculator

{

public:

void showChoice()

{

cout<<"Press '+' for addition"<<endl;

cout<<"Press '-' for subtraction"<<endl;

cout<<"Press '\*' for multiply"<<endl;

cout<<"Press '/' for divide"<<endl;

}

void add(double a, double b)

{

cout<<"Addition result: "<<a+b<<endl;

}

void subtract(double a, double b)

{

cout<<"Subtraction result: "<<a-b<<endl;

}

void multiply(double a, double b)

{

cout<<"Multiply result: "<<a\*b<<endl;

}

void divide(double a, double b)

{

cout<<"Divide result: "<<a/b<<endl;

}

};

int main()

{

Calculator obj;

obj.showChoice();

char ch;

X:

cout<<"Enter your choice"<<endl;

cin>>ch;

if(ch=='+'||ch=='-'||ch=='\*'||ch=='/')

{

cout<<"Enter two value"<<endl;

double a,b;

cin>>a>>b;

switch(ch)

{

case '+':

obj.add(a,b);

break;

case '-':

obj.subtract(a,b);

break;

case '\*':

obj.multiply(a,b);

break;

case '/':

obj.divide(a,b);

break;

}

}

else

{

cout<<"Enter Right choice"<<endl;

goto X;

}

return 0;

}

4.

#include <iostream>

using namespace std;

class Distance

{

int feet;

float inches;

public:

void setDistance(int f, float i)

{

feet =f;

inches =i;

}

void disp()

{

cout<<"Feet: "<<feet<<", Inch: "<<inches<<endl;

}

void add(Distance d1, Distance d2)

{

feet=d1.feet+d2.feet;

inches = d1.inches +d2.inches;

}

};

int main()

{

Distance obj1, obj2, obj3;

int a,c;

float b,d;

cout<<"Enter Feet and Inch for object 1"<<endl;

cin>>a>>b;

cout<<"Enter Feet and Inch for object 2"<<endl;

cin>>c>>d;

obj1.setDistance(a, b);

cout<<"Distance for object 1"<<endl;

obj1.disp();

obj2.setDistance(c,d);

cout<<"Distance for object 2"<<endl;

obj2.disp();

obj2.add(obj1,obj2);

obj3 = obj2;

cout<<"Distance for object 3"<<endl;

obj3.disp();

return 0;

}

5.

#include <iostream>

using namespace std;

class time

{

int hour,minute;

public:

void settime(int h, int m)

{

hour = h;

minute = m;

}

void showtime()

{

cout<<"Hour: "<<hour<<", Minute: "<<minute<<endl;

}

void sum(time d1, time d2)

{

hour=d1.hour+d2.hour;

minute = d1.minute +d2.minute;

}

};

int main()

{

time obj1, obj2, obj3;

int a,c, b,d;

cout<<"Enter Hour and Minute for object 1"<<endl;

cin>>a>>b;

cout<<"Enter Hour and Minute for object 1"<<endl;

cin>>c>>d;

obj1.settime(a, b);

cout<<"Time for object 1"<<endl;

obj1.showtime();

obj2.settime(c,d);

cout<<"Time for object 2"<<endl;

obj2.showtime();

obj2.sum(obj1,obj2);

obj3=obj2;

cout<<"Time for object 3"<<endl;

obj3.showtime();

return 0;

}

***Lab Work 2***

1.

#include <iostream>

using namespace std;

class A;

class student

{

int student\_number = 142014029;

char student\_name[50] = "AL-AKHIR NAYAN";

int six\_course\_marks[6]= {60,80,55,66,75,93};

public:

friend class A;

};

class A

{

public:

void calculate(student s)

{

cout<<"Student details: "<<endl;

cout<<"ID: "<<s.student\_number<<endl;

cout<<"Name: "<<s.student\_name<<endl;

double sum=0;

for(int i=0; i<6; i++)

{

sum+=s.six\_course\_marks[i];

}

cout<<"Total Marks: "<<sum<<endl;

cout<<"Average Marks: "<<sum/6<<endl;

}

};

int main()

{

A obj;

student s;

obj.calculate(s);

return 0;

}

2.

#include <iostream>

using namespace std;

class RefBooks;

class TextBooks

{

int book\_numbers;

double book\_price[5000];

public:

friend class RefBooks;

};

class RefBooks

{

public:

void calculate(TextBooks s)

{

cout<<"Insert number of books"<<endl;

cin>>s.book\_numbers;

cout<<"Insert books value"<<endl;

for(int i=0; i<s.book\_numbers; i++)

{

cin>>s.book\_price[i];

}

double sum=0;

for(int i=0; i<s.book\_numbers; i++)

{

sum+=s.book\_price[i];

}

cout<<"Total Cost: "<<sum<<" TAKA"<<endl;

}

};

int main()

{

RefBooks obj;

TextBooks s;

obj.calculate(s);

return 0;

}

3.

#include <iostream>

using namespace std;

class master;

class account;

class admin;

class person

{

int id;

char name[20];

public:

void virtual getdata()

{

cout <<"Include name and id of the employee: "<< endl;

cin >> name>>id;

}

void virtual display()

{

cout <<"\n Employee Name = "<< name<<"\n ID = "<<id;

}

void virtual bonus()

{

cout <<"\n Bonus function called (base class)";

}

friend class admin;

friend class account;

friend class master;

};

class admin : public virtual person

{

int sale\_admin;

public:

void getdata()

{

cout <<"\nSale of the admin : "<< endl;

cin >> sale\_admin;

}

void display()

{

cout <<"\n Admin Employee sale amount = "<< sale\_admin;

}

void bonus()

{

if (sale\_admin >2000)

cout << "\n The bonus of the admin is : "<< (sale\_admin\*0.25);

}

friend class master;

};

class account : public virtual person

{

long int amt;

public :

void getdata()

{

cout <<"\n Enter the amount accumulated in the account : "<< endl;

cin >> amt;

}

void display()

{

cout <<"\n Employee Account amount = "<< amt;

}

void bonus()

{

if (amt >10000)

cout << "\n The bonus of the Admin is : "<< (amt\*0.25);

}

friend class master;

};

class master : public admin, public account

{

public:

void getdata()

{

cout <<"\n Executing get data of master class :";

person::getdata();

admin:: getdata();

account::getdata();

}

void display()

{

cout <<"\n executing display of master class :";

person::display();

admin:: display();

account::display();

}

void bonus()

{

cout <<"\n executing bonus of master class :";

person::bonus();

admin:: bonus();

account::bonus();

}

};

int main()

{

person p, \*pp;

admin a, \*aa;

account ac, \*acc;

master ms, \*mt;

cout <<"\nAssigning the address of base class person to base class pointer ";

pp=&p;

pp->getdata();

pp->display();

pp->bonus();

cout <<"\nAssigning the address of admin class to base class pointer ";

pp=&a;

pp->getdata();

pp->display();

pp->bonus();

cout <<"\nAssigning the address of account class to base class pointer ";

pp=&ac;

pp->getdata();

pp->display();

pp->bonus();

cout <<"\nAssigning the address of class master to base class pointer ";

pp=&ms;

pp->getdata();

pp->display();

pp->bonus();

return 0;

}

4.

#include <iostream>

using namespace std;

class B;

class A

{

double num1;

public:

void setA(double a)

{

num1=a;

}

friend void addition(A a, B b);

};

class B

{

double num2,num3;

public:

void setB(double c, double d)

{

num2=c;

num3 =d;

}

friend void addition(A a, B b);

};

void addition(A a, B b)

{

double sum = a.num1 + b.num2+ b.num3;

cout<<"Addition result is: " <<sum<<endl;

}

int main()

{

double a,b,c;

cout<<"Insert one integer value for class A"<<endl;

cin>>a;

cout<<"Insert tow integer value for class B"<<endl;

cin>>b>>c;

A obj;

obj.setA(a);

B obj2;

obj2.setB(b,c);

addition(obj,obj2);

return 0;

}

***Lab Work 2***

***Friend Function***

2.1

#include <iostream>

using namespace std;

class time

{

public:

friend void calculation();

};

void calculation()

{

int a, b,c,d,e,f;

cout<<"Enter time T1 Hour"<<endl;

cin>>a;

cout<<"Enter time T1 Minute"<<endl;

cin>>b;

cout<<"Enter time T1 Second"<<endl;

cin>>c;

cout<<"Enter time T2 Hour"<<endl;

cin>>d;

cout<<"Enter time T2 Minute"<<endl;

cin>>e;

cout<<"Enter time T2 Second"<<endl;

cin>>f;

cout<<endl;

cout<<"T1 "<<a<<":"<<b<<":"<<c<<endl;

cout<<"T2 "<<d<<":"<<e<<":"<<f<<endl;

if (a>24&&d>24 && b>59&&c>59&&e>59&&f>59 && a<0&&d<0 && b<0&&c<0&&e<0&&f<0)

{

cout<<"Hour can not be greater than 24 and cannot be smaller than 0 and"<<endl;

cout<<"Time and Second can not be smaller than 0 and cannot be greater than 59 and"<<endl;

}

else if(a>d)

{

cout<<"T1 is greater"<<endl;

}

else if(a<d)

{

cout<<"T2 is greater"<<endl;

}

else if(a=d && b>e )

{

cout<<"T1 is greater"<<endl;

}

else if(a=d && b<e)

{

cout<<"T2 is greater"<<endl;

}

else if(a=d && b==e && c>f)

{

cout<<"T1 is greater"<<endl;

}

else if(a=d && b==e && c<f)

{

cout<<"T2 is greater"<<endl;

}

else

cout<<"Time same"<<endl;

}

int main()

{

time obj;

calculation();

return 0;

}

2.2

#include <iostream>

#include <cstring>

using namespace std;

class find\_palindrome

{

char str[30];

int check\_palindrome(char \*)

{

int length,i;

length=strlen(str);

for(i=0; i< length/2; i++)

{

if(str[i]!=str[length-1-i])

{

return 0;

}

return 1;

}

}

public:

friend void palindrome(find\_palindrome obj);

};

void palindrome(find\_palindrome obj)

{

cout<<"Enter a String: ";

cin>>obj.str;

if(obj.check\_palindrome(obj.str))

cout<<obj.str<<" is a Palindrome\n";

else

cout<<obj.str<<" is not a Palindrome\n";

}

int main()

{

find\_palindrome obj;

palindrome(obj);

return 0;

}

2.3

#include <iostream>

#include<string.h>

using namespace std;

class employee

{

protected:

int emp\_id,basic\_salary;

char emp\_name[20];

public:

void show()

{

cout<<"\nEmp. Id: "<<emp\_id<<"\n";

cout<<"Emp. Name: "<<emp\_name<<"\n";

cout<<"Basic Salary: "<<basic\_salary<<"\n";

}

};

class grossnet: public employee

{

float deduct,gross,net;

public:

grossnet(int i,char \*s, int b, float d,float g,float n)

{

emp\_id = i;

strcpy(emp\_name,s);

basic\_salary =b;

deduct =d;

gross=g;

net=n;

}

friend void calculate(grossnet g);

};

void calculate(grossnet g)

{

float deduct\_percentage,hra;

deduct\_percentage=0.4\*g.basic\_salary;

hra=0.2\*g.basic\_salary;

g.gross=g.basic\_salary+deduct\_percentage+hra;

g.net=g.gross-g.deduct;

cout<<"Gross Pay: "<<g.gross<<"\n";

cout<<"Net Pay: "<<g.net<<"\n";

}

int main()

{

int eid,ebasic;

float ededuct;

char ename[20];

cout<<"Enter:\n";

cout<<"Employee Id: ";

cin>>eid;

cout<<"Employee Name: ";

cin>>ename;

cout<<"Basic Salary: ";

cin>>ebasic;

cout<<"Deductions: ";

cin>>ededuct;

grossnet gn(eid,ename,ebasic,ededuct,0,0);

gn.show();

calculate(gn);

}

2.4

#include<iostream>

using namespace std;

class matrix

{

int a[2][2], b[2][2], ans[2][2];

public:

matrix()

{

cout << "Enter data for first matrix" << endl;

for(int i = 0; i < 2; i++)

for(int j = 0; j < 2; j++)

cin >> a[i][j];

cout << "Enter data for second matrix" << endl;

for(int i = 0; i < 2; i++)

for(int j = 0; j < 2; j++)

cin >> b[i][j];

}

friend void multiplication(matrix m);

};

void multiplication(matrix m)

{

cout << "After 2d matrix multiplication" << endl;

for(int i = 0; i < 2; i++)

{

for (int j = 0; j < 2; j++)

{

m.ans[i][j] = 0;

for(int k = 0; k < 2; k++)

m.ans[i][j] += m.a[i][k] \* m.b[k][j];

cout << m.ans[i][j] << "\t";

}

cout << endl;

}

}

int main()

{

cout << "2D Matrix Multiplication" << endl;

matrix mtMul = matrix();

multiplication(mtMul);

return 0;

}

***Lab Work 3***

1.

#include <iostream>

using namespace std;

class Student

{

public:

int age,roll;

string name;

Student(int r, string n, int a)

{

roll =r;

name =n;

age =a;

}

void display()

{

cout << "ID: " << roll << endl;

cout << "Name: " << name << endl;

cout << "Age: " << age << endl;

}

};

class ForeignStudent : public Student

{

string country ;

public:

void fdisplay( string c)

{

cout << "ID: " << roll << endl;

cout << "Name: " << name << endl;

cout << "Age: " << age << endl;

cout << "Country: " <<c<< endl;

}

};

int main()

{

Student obj(111,"Nayan",22);

obj.display();

ForeignStudent ob;

ob.display();

ob.fdisplay("Australia");

return 0;

}

2.

#include <iostream>

using namespace std;

class student

{

protected:

int roll;

char name[20];

public:

void read()

{

cout<<"Enter Roll number and name of student"<<endl;

cin>>roll;

cin.ignore();

cin.getline(name,20);

}

void display()

{

cout<<"Roll number: "<<roll<<endl;

cout<<"Student name: "<<name<<endl;

}

};

class exam: public student

{

protected:

double marks[3];

public:

void read\_marks()

{

cout<<"Insert marks of 3 subjects"<<endl;

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

{

cin>>marks[i];

}

}

void display\_marks()

{

cout<<"Marks of 3 subjects"<<endl;

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

{

cout<<marks[i]<<endl;

}

}

};

class result: public exam

{

protected:

double total;

public:

void display\_result()

{

cout<<"Roll number: "<<roll<<endl;

cout<<"Student name: "<<name<<endl;

cout<<"Marks of 3 subjects"<<endl;

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

{

cout<<marks[i]<<endl;

}

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

{

total+=marks[i];

}

cout<<"Total marks: "<<total<<endl;

}

};

int main()

{

result obj;

obj.read();

obj.read\_marks();

obj.display\_result();

return 0;

}

3.

#include <iostream>

using namespace std;

class staff

{

protected:

int staff\_id;

char level[20];

public:

void read()

{

cout<<"Enter Staff Id and level"<<endl;

cin>>staff\_id;

cin.ignore();

cin.getline(level,20);

}

void display\_staff()

{

cout<<"Staff Details: "<<endl;

cout<<"Staff ID : "<<staff\_id<<endl;

cout<<"Level: "<<level<<endl;

}

};

class teacher

{

protected:

int teacher\_id;

char subject[20];

public:

void read\_teacher()

{

cout<<"Insert Teacher id and subject"<<endl;

cin>>teacher\_id;

cin.ignore();

cin.getline(subject,20);

}

void display\_teacher()

{

cout<<"Teacher Details: "<<endl;

cout<<"Teacher ID : "<<teacher\_id<<endl;

cout<<"Subject: "<<subject<<endl;

}

};

class coordinator: public teacher, public staff

{

protected:

char program[30];

public:

void read\_coordinator()

{

cout<<"Insert Program of Coordinator"<<endl;

cin.ignore();

cin.getline(program,20);

}

void display\_coordinator()

{

cout<<"Coordinator Details: "<<endl;

cout<<"Coordinator: "<<program<<endl;

}

};

int main()

{

coordinator obj;

obj.read();

obj.read\_teacher();

obj.read\_coordinator();

obj.display\_staff();

obj.display\_teacher();

obj.display\_coordinator();

return 0;

}

4.

#include <iostream>

using namespace std;

class student

{

protected:

int roll;

char name[20];

public:

void read()

{

cout<<"Enter Roll number and name of student"<<endl;

cin>>roll;

cin.ignore();

cin.getline(name,20);

}

void display()

{

cout<<"Roll number: "<<roll<<endl;

cout<<"Student name: "<<name<<endl;

}

};

class exam: public student

{

protected:

double marks[3];

public:

void read\_marks()

{

cout<<"Insert marks of 3 subjects"<<endl;

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

{

cin>>marks[i];

}

}

void display\_marks()

{

cout<<"Marks of 3 subjects"<<endl;

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

{

cout<<marks[i]<<endl;

}

}

};

class sport

{

protected:

int score;

public:

void read\_sport()

{

cout<<"Insert score of student"<<endl;

cin>>score;

}

void display\_score()

{

cout<<"Student Sport Score: "<<score<<endl;

}

};

class result: public exam, public sport

{

protected:

double total;

double average;

public:

void display\_result()

{

cout<<"Roll number: "<<roll<<endl;

cout<<"Student name: "<<name<<endl;

cout<<"Marks of 3 subjects"<<endl;

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

{

cout<<marks[i]<<endl;

}

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

{

total+=marks[i];

}

cout<<endl;

}

void calculate()

{

average = total/3;

cout<<"Total marks: "<<total<<endl;

cout<<"\nAverage marks: "<<average<<endl;

}

};

int main()

{

result obj;

obj.read();

obj.display();

obj.read\_marks();

obj.display\_marks();

obj.read\_sport();

obj.display\_result();

obj.display\_score();

obj.calculate();

return 0;

}

5.

#include <iostream>

using namespace std;

class Publication

{

protected:

char title[200];

double price;

public:

virtual void getdata()

{

cout<<"Insert Publication title and price"<<endl;

cin.getline(title,200);

cin.ignore();

cin>>price;

}

virtual void putdata()

{

cout<<"Publication title: "<<title<<endl;

cout <<"Publication price: "<<price<<endl;

}

};

class book: public Publication

{

protected:

int page\_count;

public:

void getdata()

{

cout<<"Insert Book Page Count"<<endl;

cin>>page\_count;

}

void putdata()

{

cout<<"Page Count: "<<page\_count<<endl;

}

};

class tape: public Publication

{

protected:

int tape;

public:

void getdata()

{

cout<<"Insert Tape"<<endl;

cin>>tape;

}

void putdata()

{

cout <<"Tape: "<<tape<<" Minutes"<<endl;

}

};

int main()

{

Publication p, \*pt;

tape o, \*ob;

book oo, \*obj;

pt=&p;

pt->getdata();

pt->putdata();

pt=&o;

pt->getdata();

pt->putdata();

pt=&oo;

pt->getdata();

pt->putdata();

return 0;

}

***Lab Work 4***

1.

#include <iostream>

using namespace std;

class SalaryAccount

{

int Emp\_id;

string Emp\_name;

string Designation;

double Basic\_pay;

double House\_allowance, Medical\_allowance, Transport\_allowance;

public:

SalaryAccount()

{

Emp\_id= 101;

Emp\_name = "Nazim Uddin";

Designation ="Designation Manager";

Basic\_pay = 15000;

}

SalaryAccount(int a, char b[], char c[], double d)

{

Emp\_id= a;

Emp\_name = b;

Designation =c;

Basic\_pay = d;

}

void Calculate()

{

House\_allowance = (Basic\_pay\*30)/100;

Medical\_allowance = (Basic\_pay\*20)/100;

Transport\_allowance = (Basic\_pay\*20)/100;

}

void Display(){

cout<<"Employ Id: "<<Emp\_id<<endl;

cout<<"Employ Name : "<<Emp\_name<<endl;

cout<<"Employ Designation: "<<Designation<<endl;

cout<<"Employ Basic Pay: "<<Basic\_pay<<" TAKA"<<endl;

cout<<"Employ House Allowance: "<<House\_allowance<<" TAKA"<<endl;

cout<<"Employ Medical Allowance: "<<Medical\_allowance<<" TAKA"<<endl;

cout<<"Employ Transport Allowance: "<<Transport\_allowance<<" TAKA"<<endl;

cout<<"Total Salary : "<<Basic\_pay-(Medical\_allowance+ House\_allowance+Transport\_allowance)<<" TAKA"<<endl;

cout<<endl;

}

};

int main()

{

char name[20],designation[20];

double basic\_pay;

int id;

cout<<"Insert Employ Id"<<endl;

cin>>id;

cout<<"\nInsert Employ Name"<<endl;

cin.ignore();

cin.getline(name,20);

cout<<"\nInsert Employ Designation"<<endl;

cin.getline(designation,20);

cout<<"\nInsert Employ Basic Pay"<<endl;

cin>>basic\_pay;

SalaryAccount emp1, emp2(id,name,designation,basic\_pay);

emp1.Calculate();

emp1.Display();

emp2.Calculate();

emp2.Display();

return 0;

2.

#include <iostream>

#define pi 3.1416

using namespace std;

class Polygon

{

public:

void calculate(double r){

cout<<"The area of Circle is: "<<2\*pi\*r<<" meter"<<endl;

}

void calculate(float s){

cout<<"The area of Square is: "<<s\*s<<" meter"<<endl;

}

void calculate(double a, double b){

cout<<"The area of Triangle is: "<<.5\*a\*b<<" meter"<<endl;

}

};

int main()

{

double value1;

float value2;

double value3,value4;

cout<<"Insert radius of circle"<<endl;

cin>>value1;

cout<<"\nInsert one value for Square"<<endl;

cin>>value2;

cout<<"\nInsert value for triangle ground and height"<<endl;

cin>>value3>>value4;

Polygon obj;

obj.calculate(value1);

obj.calculate(value2);

obj.calculate(value3,value4);

return 0;

}