Assignment 32

1. Create a class FLOAT that contains one float data member. Overload all the four

arithmetic operators so that they can operate on the objects of FLOAT.

#include<iostream>

using namespace std;

class Float

{

    private:

        int x;

    public:

        Float()

        {

        }

        Float(int m)

        {

            x=m;

        }

        void set(int m)

        {

            x=m;

        }

        float get()

        {

            return x;

        }

        Float operator+(Float f)

        {

            Float temp;

            temp.x=x+f.x;

            return temp;

        }

        Float operator-(Float f)

        {

            Float temp;

            temp.x= x - f.x;

            return temp;

        }

        Float operator\*(Float f)

        {

            Float temp;

            temp.x = x \* f.x;

            return temp;

        }

        Float operator/(Float f)

        {

            Float temp;

            temp.x= x / f.x;

            return temp;

        }

};

int main()

{

    Float c1,c2,c3,c4,c5,c6;

    c1.set(4);

    c2.set(5);

    c3=c1+c2;

    cout<<"Addition of c1 and c2 is     : ";

    cout<<c3.get();

    c4.set(9);

    c5.set(5);

    c6=c5-c4;

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

    cout<<"subtraction of the two numbers is    : ";

    cout<<c5.get();

    c5=c1 \* c3;

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

    cout<<" multiplication of the numbers is     ";

    cout<<c5.get();

    c5=c2/c4;

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

    cout<<"division of the numbers is    ";

    cout<<c5.get();

    return 0;

}

Output:

cd "c:\Users\tusha\Documents\coadind\assignment\_32\" ; if ($?) { g++ float.cpp -o float } ; if ($?) { .\float }

Addition of c1 and c2 is : 9

----------------------------

subtraction of the two numbers is : 5

-----------------------------

multiplication of the numbers is 36

-------------------------------

division of the numbers is 0

PS C:\Users\tusha\Documents\coadind\assignment\_32>onst

2. Define a class Rectangle and overload area function for different types of data type.

#include<iostream>

using namespace std;

class Rectangle

{

    float area;

    int x;

    int y;

    public:

        Rectangle()

        {

        }

        void display()

        {

            cout<<" area of the reactangle is : "<<area<<" sq. cm"<<endl;

        }

        Rectangle(int a, int b)

        {

            x=a;

            y=b;

        }

        Rectangle(int a, float b)

        {

            x=a;

            y=b;

        }

        Rectangle(float a, float b)

        {

            x=a;

            y=b;

        }

        Rectangle(double a, double b)

        {

            x=a;

            y=b;

        }

        Rectangle(double a, int b)

        {

            x=a;

            y=b;

        }

        Rectangle(int a, double b)

        {

            x=a;

            y=b;

        }

        void cal\_area()

        {

            area=x\*y;

        }

};

int main()

{

    Rectangle r(4,5);

    r.cal\_area();

    r.display();

    Rectangle r1(4.0,5.3);

    r1.cal\_area();

    r1.display();

    Rectangle r2(4.5, 3);

    r2.cal\_area();

    r2.display();

    Rectangle r3(4.5,3.5);

    r3.cal\_area();

    r3.display();

    return 0;

}

d "c:\Users\tusha\Documents\coadind\assignment\_32\" ; if ($?) { g++ different\_data\_tyoe.cpp -o different\_data\_tyoe } ; if ($?) { .\different\_data\_tyoe }

area of the reactangle is : 20 sq. cm

area of the reactangle is : 20 sq. cm

area of the reactangle is : 12 sq. cm

area of the reactangle is : 12 sq. cm

PS C:\Users\tusha\Documents\coadind\assignment\_32>

3. Define a base class Animals having member function sound() . Define another

derived class from Animals class named Dogs. You need to override the sound

function of the base class in the derived class.

#include<iostream>

using namespace std;

class Animal

{

    protected:

        string sound;

    public:

        string Sound()

        {

            return sound;

        }

        void set\_sound(string sound)

        {

            this->sound=sound;

        }

};

class dog : public Animal

{

    private:

        string dog\_sound;

    public:

        string Sound()

        {

            return dog\_sound;

        }

        void set\_sound(string sound)

        {

            this->dog\_sound= sound;

        }

};

int main()

{

    Animal cat;

    cat.set\_sound("Maw Maw");

    cout<<"the sound of the cat is "<<cat.Sound()<<endl;

    dog pet;

    pet.set\_sound("Bow Bow");

    cout<<"the xound of the dog is "<<pet.Sound()<<endl;

    return 0;

}

Ou[ut:

> cd "c:\Users\tusha\Documents\coadind\assignment\_32\" ; if ($?) { g++ problem3.cpp -o problem3 } ; if ($?) { .\problem3 }

the sound of the cat is Maw Maw

the xound of the dog is Bow Bow

PS C:\Users\tusha\Documents\coadind\assignment\_32>

4. Define a class Addition that can add 2 or 3 numbers of different data types using

function overloading.

5. Define a class A having multiple constructors. Define another class B derived from

class A. Create derived class constructors and show use of constructor in this single

inheritance.

#include<iostream>

using namespace std;

class person

{

    protected:

        string name;

        int age;

    public:

        person()

        {

            cout<<"default constructpr is called ..."<<endl;

        }

        person(string name, int age)

        {

            this ->name= name;

            this->age= age;

        }

};

class student : public person

{

    int roll\_no;

    public:

        student():person()

        {

        }

        student(long roll\_no, string name, int age) : person(name, age)

        {

            this->roll\_no= roll\_no;

        }

        void display()

        {

            cout<<"Roll No.     : "<<roll\_no<<endl;

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

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

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

        }

};

int main()

{

    student s(1751059, "Tushar Maliye", 23);

    s.display();

    return 0;

}

Output:

cd "c:\Users\tusha\Documents\coadind\assignment\_32\" ; if ($?) { g++ problem\_05.cpp -o problem\_05 } ; if ($?) { .\problem\_05 }

Roll No. : 1751059

Name : Tushar Maliye

Age : 23

-------------------

PS C:\Users\tusha\Documents\coadind\assignment\_32>

6. C++ Program to illustrate the use of Constructors in multilevel inheritance of your

choice.

#include<iostream>

using namespace std;

class factorial

{

    private:

        int x, fact;

    public:

        factorial()

        {

            fact=1;

        }

        factorial(int n)

        {

            x=n;

            fact=1;

        }

        factorial(factorial &f)// copy constructor

        {

            x = f.x;

            fact=1;

        }

        void calculate()

        {

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

            {

                fact=fact\*i;

            }

        }

        void display()

        {

            cout<<"\n the factorial is "<<fact<<endl;

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

        }

};

int main()

{

    factorial f(5);

    f.calculate();

    f.display();

    factorial f1(f);

    f1.calculate();

    f1.display();

    return 0;

}

Output:

cd "c:\Users\tusha\Documents\coadind\assignment\_32\" ; if ($?) { g++ factorial.cpp -o factorial } ; if ($?) { .\factorial }

the factorial is 120

-------------------

the factorial is 120

-------------------

PS C:\Users\tusha\Documents\coadind\assignment\_32>

10. Create a C++ class for player objects with the following attributes: player no., name,

number of matches and number of goals done in each match. The number of

matches varies for each player. Write a parameterized constructor which initializes

player no., name, number of subjects and creates an array for number of goals and

number of matches dynamically.

#include<iostream>

using namespace std;

class player

{

        string name;

        long long phone\_no;

        int no\_match;

        int \*no\_goals;

        int count;

    public:

        player()

        {

            count=0;

        }

        void accept\_details()

        {

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

            cout<<"enter the namw           : ";

            cin>>name;

            cout<<"enter the phone no       : ";

            cin>>phone\_no;

            cout<<"enter the no. of matches : ";

            cin>>no\_match;

            no\_goals= new int[4];

            for(int i=1; i<=no\_match; i++)

            {

                cout<<"enter goals in "<<i<<" match : ";

                cin>>no\_goals[i];

            }

        }

        void display()

        {

            cout<<"---------player data-----------"<<endl;

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

            cout<<"phone no          : "<<phone\_no<<endl;

            cout<<"No. of matches    : "<<no\_match;

            for (int i = 1; i <= no\_match; i++)

            {

                cout<<"\nGoals in "<<i<<" match : "<<no\_goals[i];

                count=count + no\_goals[i];

            }

            cout<<"\ntotal goals       : "<<count<<endl;

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

        }

};

int main()

{

     player p;

     p.accept\_details();

     p.display();

     return 0;

}

Output:

cd "c:\Users\tusha\Documents\coadind\assignment\_32\" ; if ($?) { g++ problem10.cpp -o problem10 } ; if ($?) { .\problem10 }

-------------------------------

enter the namw : tushar

enter the phone no : 7798018146

enter the no. of matches : 4

enter goals in 1 match : 5

enter goals in 2 match : 6

enter goals in 3 match : 4

enter goals in 4 match : 2

---------player data-----------

Name : tushar

phone no : 7798018146

No. of matches : 4

Goals in 1 match : 5

Goals in 2 match : 6

Goals in 3 match : 4

Goals in 4 match : 2

total goals : 17

------------------------------

PS C:\Users\tusha\Documents\coadind\assignment\_32>