**Assignment - 26 A Job Ready Bootcamp in C++, DSA and IOT MySirG**

**Member function, static, constructor**

**Abhishek Kumar**

// Q.1.

#include<iostream>

using namespace std;

class Complex

{

    private:

       int a,b;

    public:

    void setData(int x, int y)

    {

        a = x;

        b = y;

    }

    void showData()

    {

        cout << "real = " << a << " Img = " << b << endl;

    }

    Complex add(Complex c)

    {

        Complex temp;

        temp.a = a+c.a;

        temp.b = b+c.b;

        return temp;

    }

};

int main()

{

    Complex c1,c2,c3;

    c1.setData(5,4);

    c2.setData(6,4);

    c3 = c1.add(c2);

    c3.showData();

}

// Q.2.

#include<iostream>

using namespace std;

class Time

{

    private:

       int hour;

       int min;

       int sec;

    public:

    void setTime(int x, int y, int z)

    {

        hour = x;

        min = y;

        sec = z;

    }

    void normarlize()

    {

        min = min+sec/60;

        sec = sec%60;

        hour = hour+min/60;

        min = min%60;

    }

    Time add(Time t)

    {

        Time temp;

        temp.sec = sec+t.sec;

        temp.min = min+t.min;

        temp.hour = hour+t.hour;

        temp.normarlize();

        return (temp);

    }

    void showTime()

    {

        cout << endl;

        cout << hour << " : " << min << " : " << sec ;

    }

};

int main()

{

    Time t1,t2,t3;

    t1.setTime(4, 5, 8);

    t2.setTime(2, 3, 1);

    t3 = t1.add(t2);

    t1.showTime();

    t2.showTime();

    cout << endl;

    t3.showTime();

}

// Q.3.

#include<iostream>

using namespace std;

class Cube

{

    public:

      double side;

    double volume()

    {

        return (side\*side\*side);

    }

    Cube (double side1)

    {

        cout << "A constructor is called:"<< endl;

        side = side1;

    }

    Cube()

    {

        cout << "A default consturctor is called:"<< endl;

    }

    ~Cube()

    {

        cout << "Destructing "<< side << endl;

    }

};

int main()

{

    Cube c1(2.34);

    Cube c2;

    cout << "The side of the cube: "<< c1.side << endl;

    cout << "The volume of first cube is: "<< c1.volume() << endl;

    cout << "Enter the length of the second cube:";

    cin >> c2.side;

    cout << "The volum of second cube is: "<<c2.volume()<< endl;

    return 0;

}

// Q.4.

#include<iostream>

using namespace std;

class counter

{

    private:

       int count;

    public:

    counter()

    {

        count=0;

    }

    void inc\_count()

    {

        count++;

    }

    int get\_count()

    {

        return count;

    }

};

int main()

{

    counter c1;

    cout << "\nBefore call counter function count = ";

    cout << c1.get\_count();

    c1.inc\_count();

    cout << "\nAfter call counter function count = ";

    cout << c1.get\_count();

    return 0;

}

// Q.5.

#include<iostream>

using namespace std;

class Date

{

    private:

       int dd,mm,yyyy;

    public:

    Date()

    {

        dd = 19;

        mm = 10;

        yyyy = 2020;

    }

    void display()

    {

        cout << "Today Date is ......."<< endl;

        cout << dd << "-" << mm << "-" << yyyy << endl;

    }

};

int main()

{

    Date d;

    d.display();

}

// Q.6.

#include<iostream>

using namespace std;

class student

{

    private:

       int mark;

       char grade;

    public:

    student(int m, int g)

    {

        mark = m;

        grade = g;

    }

    void display()

    {

        cout << "Mark is " << mark <<" Grade " << grade << endl;

    }

};

int main()

{

    student s1(320 , 'B'),s2(435, 'A');

    cout << "Record of the student is ::..............."<< endl;

    s1.display();

    cout << "Record of the student is ::................"<< endl;

    s2.display();

}

// Q.7.

#include<iostream>

using namespace std;

class Box

{

    public:

        double lenght,breath,height;

    double volume()

    {

        cout << "Lenght of the Box: " << lenght << endl;

        cout << "Breath of the Box: " << breath << endl;

        cout << "Height of the Box: " << height << endl;

        return (lenght\*breath\*height);

    }

    Box (double l, double b, double h)

    {

        lenght = l;

        breath = b;

        height = h;

    }

    Box()

    {

        cout << "\n A Copy Consturctor Called "<< endl;

    }

    ~Box()

    {

        cout << "\n A Default Consturctor Called "<< endl;

    }

};

int main()

{

    Box b1(2.4,5.7,2.1),b2(3.3,4.4,5.5);

    cout << "Volume of the Box: " << b1.volume() << endl;

    cout << endl;

    cout << "Volume of the Box: " << b2.volume() << endl;

}

// Q.8.

#include<iostream>

using namespace std;

class Bank

{

    public:

        float principla,rate,year,si;

    float setData()

    {

        return si = (principla\*rate\*year)/100;

    }

    Bank(float p, float r, float y)

    {

        principla = p;

            rate = r;

            year = y;

    }

};

int main()

{

    Bank b(1200,2,5.4);

    cout << "Simple Interest " << b.setData() << endl;

}

// Q.10.

#include<iostream>

using namespace std;

class counter

{

    private:

       static int count;

    public:

    counter()

    {

        count=0;

    }

    void inc\_count()

    {

        count++;

    }

    int get\_count()

    {

        return count;

    }

};

int counter::count;

int main()

{

    counter c1;

    cout << c1.get\_count();

    return 0;

}