CPP ASSIGNMENT-04 (26)

**1. Define a class Complex to represent a complex number with instance variables a and b to store real and imaginary parts. Also define following member functions**

**a. void setData (int, int)**

**b. void showData ()**

**c. Complex add (Complex)**

#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 part = "<<a<<" Imaginary part = "<<b<<endl;

        }

        Complex add (Complex c)

        {

            Complex temp;

            temp. a = a + c. a;

            temp. b = b + c. b;

            return temp;

        }

};

int main ()

{

    int m, n;

    Complex c1, c2, c3;

    c1. setData (8, 12);

    c2. setData (2, 8);

    c3 = c1.add (c2);

    c3. showData ();

    return 0;

}

**2. Define a class Time to represent a time with instance variables h, m and s to store hour, minute and second. Also define following member functions**

**a. void setTime (int, int, int)**

**b. void showTime ()**

**c. void normalize ()**

**d. Time add (Time)**

#include<iostream>

using namespace std;

class Time

{

    private:

        int h, m, s;

    public:

        Time () {h=0; m=0; s=0;}

        void SetTime (int x, int y, int z)

          {h=x; m=y; s=z;}

        void ShowTime ()

          {cout<<h<<" Hour "<<m<<" Minute "<<s<<" Second"<<endl;}

        void normalize ()

        {

            h = h + m/60;

            m = (m%60) + s/60;

            s = s%60;

            // cout<<h<<" hour"<<m<<" minute "<<s<<" second"<<endl;

        }

        Time add (Time x)

        {

            Time tim;

            tim.h = h + x.h;

            tim.m = m + x.m;

            tim.s = s + x.s;

            return tim;

        }

};

int main ()

{

    Time T1, T2, T3;

    T1.SetTime (2, 40, 60);

    T2.SetTime (4, 50, 50);

    T3 = T1.add(T2);

    T3.normalize ();

    T3.ShowTime ();

    return 0;

}

**3. Define a class Cube and calculate Volume of Cube and initialise it using constructor.**

#include<iostream>

using namespace std;

class Cube

{

    private:

        int a;

    public:

        Cube () {a = 0;}

        void setValue (int x)

        {

            a = x;

        }

        void Volume ()

        {

            cout<<"Volume of Cube is "<< a\*a\*a;

        }

};

int main ()

{

    int m;

    cout<<"Enter side of Cube:";

    cin >> m;

    Cube C1;

    C1. SetValue(m);

    C1. Volume ();

    return 0;

}

**4. Define a class Counter and Write a program to Show Counter using Constructor.**

#include<iostream>

using namespace std;

class Counter

{

    private:

        int c;

    public:

        Counter ()

        {

             int c = 0;

             cout<< "Count: "<< ++c <<endl;

        }

};

int main ()

{

    Counter C1;

}

**5. Define a class Date and write a program to Display Date and initialise date object using Constructors.**

#include<iostream>

using namespace std;

class Date

{

    private:

        int dd, mm, yyyy;

    public:

        Date ()

        {

             dd = 00; mm = 00; yyyy = 0000;

        }

        void SetDate (int x, int y, int z)

        {

            dd = x; mm = y; yyyy = z;

        }

        void ShowDate ()

        {

            cout <<"Day: "<<dd <<" Month: "<<mm <<" Year: "<<yyyy;

        }

};

int main ()

{

    int a, b, c;

    cout<<"Enter Date: ";

    cin>>a>>b>>c;

     Date D1;

    D1. SetDate(a, b, c);

    D1. ShowDate();

    return 0;

}

**6. Define a class student and write a program to enter student details using constructor and define member function to display all the details.**

#include<iostream>

using namespace std;

class Student

{

    private:

        char name [20], branch [10];

        int Roll;

    public:

        Student ()

        {

            cout<<"Enter Name, Branch and roll no.: ";

            cin.getline (name, 20);

            cin.getline (branch, 10);

            cin>>Roll;

        }

        void showDetails ()

        {

            cout<<"Name: "<<name<<"  Branch: "<<branch<<"  Roll no: "<<Roll<<endl;

        }

};

int main ()

{

    Student s1;

    s1. showDetails ();

    return 0;

}

**7. Define a class box and write a program to enter length, breadth and height and initialise objects using constructor also define member functions to calculate volume of the box.**

#include<iostream>

using namespace std;

class Box

{

    private:

        int l, b, h;

    public:

        Box ()

        {

            l=0; b=0; h=0;

        }

    void setData (int x, int y, int z)

    {

        l = x; b = y; h = z;

    }

    void ShowVolume ()

    {

        cout<<"Volume of the Box: "<<l\*b\*h;

    }

};

int main ()

{

    int l, b, h;

    Box b1;

    cout<<"Enter length, breadth and height: ";

    cin>>l>>b>>h;

    b1.setData (l, b, h);

    b1.ShowVolume ();

    return 0;

}

**8. Define a class Bank and define member functions to read principal, rate of interest and year. Another member functions to calculate simple interest and display it. Initialise all details using constructor.**

#include <iostream>

using namespace std;

class Bank

{

private:

    int P, Y;

    float roi, SI;

public:

    Bank ()

    {

        P = 0; roi = 3.5f; Y = 0, SI =0;

        cout<<"Your rate of interest is:"<<roi<<" % p.a"<<endl;

    }

    void SetData (int x, int z)

    {

        P = x; Y = z;

    }

    void getSimpleIntrest ()

    {

        SI = (P\*roi\*Y)/100;

        cout<<"Your SI for "<<Y<<" years is Rs."<<SI<<endl;

    }

};

int main ()

{

    int p, t;

    cout<<"Enter principal value and time:"<<endl;

    cin>>p>>t;

    Bank b1;

    b1.SetData (p, t);

    b1.getSimpleIntrest ();

    return 0;

}

**9. Define a class Bill and define its member function get () to take detail of customer, calculateBill () function to calculate electricity bill using below tariff: Up to 100 units RS. 1.20 per unit from 100 to 200 units RS. 2 per unit Above 200 units RS. 3 per unit.**

#include<iostream>

using namespace std;

class Bill

{

    private:

       float unit;

    public:

    Bill ()

    {

        unit = 0;

    }

        void getDetails (float x)

        {

            unit = x;

        }

        void CalculateBill ()

        {

            if (unit <= 100)

                cout<<"Your electricity bill is Rs.:"<<unit\*1.20<<endl;

            else if (unit >= 100 && unit <= 200)

                cout<<"Your electricity bill is Rs.:"<< (120 + (unit-100) \*2) <<endl;

            else

                cout<<"Your electricity bill is Rs.:"<<((unit-200) \*3+ 320) <<endl;

        }

};

int main ()

{

    int u;

    cout <<"Enter units:";

    cin>>u;

    Bill b;

    b. getDetails(u);

    b. CalculateBill ();

    return 0;

}

**10. Define a class StaticCount and create a static variable. Increment this variable in a function and call this 3 times and display the result.**

#include<iostream>

using namespace std;

class StaticCount

{

    private:

        static int a;

    public:

        void increamentVar ()

        {

            a++;

        }

        void showVar ()

        {

            cout<<"New value is:"<<a;

        }

};

int StaticCount :: a;

int main ()

{

    StaticCount S1;

    S1.increamentVar();

    S1.increamentVar();

    S1.increamentVar();

    S1.showVar();

    return 0;

}