**Assignment-26 Solution: Name: Om Pant**

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)

ans-

// 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{

    int a,b;

    public:

        void setData(int x,int y){

            a = x;

            b = y;

        }

        void showData(){

            if(b>0)

                cout<<"Complex Number: "<<a<<" + "<<b<<"i";

            else

                cout<<"Complex Number: "<<a<<" "<<b<<"i";

        }

        Complex add(Complex x){

            Complex temp;

            temp.a = a + x.a;

            temp.b = b + x.b;

            return temp;

        }

};

int main(){

    Complex c1,c2,c3;

    int a,b;

    cout<<"Enter real part of Complex 1: ";

    cin>>a;

    cout<<"Entre imaginary part of Complex 1: ";

    cin>>b;

    c1.setData(a,b);

    cout<<"Enter real part of Complex 2: ";

    cin>>a;

    cout<<"Entre imaginary part of Complex 2: ";

    cin>>b;

    c2.setData(a,b);

    c3 = c1.add(c2);

    cout<<"Addition of Complex Numbers: "<<endl;

    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)

ans-

// 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:

        void setTime(int x,int y, int z){

            if(x>=0 && x<24){

                h = x;

            }else{

                h = -1;

            }

            if(y>=0 && y<60){

                m = y;

            }else{

                m = -1;

            }

            if(z>=0 && z<60){

                s = z;

            }else{

                s = -1;

            }

        }

        void showTime(){

            cout<<"Time: "<<h<<"hr "<<m<<"min "<<s<<"sec "<<endl;

        }

        void normalize(){

            m = m + s/60;

            s = s%60;

            h = h + m/60;

            m = m%60;

        }

        Time add(Time t){

            Time temp;

            temp.h = h + t.h;

            temp.m = m + t.m;

            temp.s = s + t.s;

            temp.normalize();

            return temp;

        }

};

int main(){

    Time t1,t2,t3;

    int x,y,z;

    cout<<"Enter Time "<<endl;

    cout<<"Enter Hours: ";

    cin>>x;

    cout<<"Enter Minutes: ";

    cin>>y;

    cout<<"Enter Seconds: ";

    cin>>z;

    t1.setTime(x,y,z);

    cout<<"Enter Time "<<endl;

    cout<<"Enter Hours: ";

    cin>>x;

    cout<<"Enter Minutes: ";

    cin>>y;

    cout<<"Enter Seconds: ";

    cin>>z;

    t2.setTime(x,y,z);

    t1.showTime();

    t2.showTime();

    t3 = t1.add(t2);

    t3.showTime();

    return 0;

}

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

Ans-

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

#include<iostream>

using namespace std;

class Cube{

    int side;

    public:

        Cube(int n){

            side = n;

        }

        int volume(){

            return side\*side\*side;

        }

};

int main(){

    Cube c(5);

    cout<<"Volume: "<<c.volume()<<endl;

    return 0;

}

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

Ans-

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

#include<iostream>

using namespace std;

class Count{

    static int count;

    public:

        Count(){

            count++;

        }

        void getCount(){

            cout<<count<<endl;

        }

};

int Count:: count;

int main(){

    Count c1,c2,c3;

    c1.getCount();

}

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

Ans-

// 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{

    int day,month,year;

    public:

        Date(int d, int m, int y){

            day = d;

            month = m;

            year = y;

        }

        void showDate(){

            cout<<"Date (DD:MM:YY) : "<<day<<"\\"<<month<<"\\"<<year<<endl;

        }

};

int main(){

    Date d(12,5,2024);

    d.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.

Ans-

// 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{

    string name;

    int rollNo;

    int age;

    public:

        Student(string s, int r, int a){

            name = s;

            rollNo = r;

            age = a;

        }

        void showDetails(){

            cout<<"Student Name: "<<name<<endl<<"Student Roll No. : "<<rollNo<<endl<<"Student Age: "<<age<<endl;

        }

};

int main(){

    string n;

    int rn,age;

    cout<<"Enter Student Name, Roll No. and Age ";

    getline(cin,n);

    cin>>rn>>age;

    Student s = Student(n,rn,age);

    s.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.

Ans-

// 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{

    int length,breadth,height;

    public:

        Box(){}

        Box(int l, int b, int h){

            length = l;

            breadth = b;

            height = h;

        }

        int volume(){

            return length\*breadth\*height;

        }

};

int main(){

    int l,b,h;

    cout<<"Enter Length, Breadth and Height of box: ";

    cin>>l>>b>>h;

    Box box1 = Box(l,b,h);

    cout<<"Volume: "<<box1.volume()<<endl;

}

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.

Ans-

// 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{

    int principal,rate,year;

    public:

        int getPrincipal(){

            return principal;

        }

        int getRate(){

            return rate;

        }

        int getYear(){

            return year;

        }

        Bank(int p, int r, int t){

            principal = p;

            rate = r;

            year = t;

        }

        float SimpleInterese(){

            return (principal \* rate \* year) / 100;

        }

};

int main(){

    int p,r,t;

    cout<<"Enter Principal, Rate and Year of Interest: ";

    cin>>p>>r>>t;

    Bank x = Bank(p,r,t);

    cout<<"Principal : "<<x.getPrincipal()<<endl;

    cout<<"Rate : "<<x.getRate()<<endl;

    cout<<"Year : "<<x.getYear()<<endl;

    cout<<"Simple Interest: "<<x.SimpleInterese()<<endl;

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 :

Upto 100 unit RS. 1.20 per unit

From 100 to 200 unit RS. 2 per unit

Above 200 units RS. 3 per unit.

Ans-

// 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 :

// Upto 100 unit RS. 1.20 per unit

// From 100 to 200 unit RS. 2 per unit

// Above 200 units RS. 3 per unit.

#include<iostream>

using namespace std;

class Bill{

    int customer\_ID;

    string cus\_name;

    int unit;

    public:

        void get(){

            cout<<"Enter Customer Details"<<endl;

            cout<<"ID: ";

            cin>>customer\_ID;

            fflush(stdin);

            cout<<"Name: ";

            getline(cin,cus\_name);

        }

        float calculateBill(){

            cout<<"Enter units Consumed: ";

            cin>>unit;

            if(unit>0 && unit<100){

                return unit\*1.20;

            }

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

                return unit\*2.0;

            }

            else if (unit>200){

                return unit\*3.0;

            }

            else{

                return -1;

            }

        }

};

int main(){

    Bill b1;

    b1.get();

    cout<<"Your Bill is: $"<<b1.calculateBill()<<endl;

    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.

Ans-

// 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{

    static int var;

    public:

        void increment(){

            var++;

        }

        void getValue(){

            cout<<var;

        }

};

int StaticCount::var;

int main(){

    StaticCount s;

    s.increment();

    s.increment();

    s.increment();

    s.getValue();

    return 0;

}