

Assignment - 26

Member function, static, constructor

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)

Program -

```
#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<<"    Imaginary = "<<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(4,5);
    c2.setData(9,3);
    c1.showData();
```

```

    c2.showData();
    c3 = c1.add(c2);
    c3.showData();
    return 0;
}

```

Output -

```

Real = 4 Imaginary = 5
Real = 9 Imaginary = 3
Real = 13 Imaginary = 8

```

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)

Program -

```

#include<iostream>
using namespace std;

class Time
{
    private:
        int h, m, s;

    public:
        void setTime(int x, int y, int z)
        {
            h = x;
            m = y;
            s = z;
        }
        void showTime()
        {
            cout<<h<<" hours "<<m<<" minutes "<<s<<" seconds"<<endl;
        }
        void normalize()
        {
            if(s > 60)
            {
                s = s - 60;
            }
        }
    };
}

```

```

        m = m + 1;
    }
    if(m > 60)
    {
        m = m - 60;
        h = h + 1;
    }
}
Time add(Time T)
{
    Time temp;
    temp.h = h + T.h;
    temp.m = m + T.m;
    temp.s = s + T.s;
    return temp;
}

};

int main()
{
    Time t1, t2, t3;
    t1.setTime(1,4,5);
    t2.setTime(3,58,60);
    t1.showTime();
    t2.showTime();
    t3 = t1.add(t2);
    t3.normalize();
    cout<<"Total time is-\n";
    t3.showTime();
    return 0;
}

```

Output -

1 hours 4 minutes 5 seconds
 3 hours 58 minutes 60 seconds
 Total time is-
 5 hours 3 minutes 5 seconds

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

Program -

```

#include<iostream>
using namespace std;

```

```

class Cube
{
    private:
        float vol;

    public:
        Cube()
        {
            vol = 0.0f;
        }
        Cube(float s)
        {
            vol = s*s*s;
        }
        void displayVolume()
        {
            cout<<vol;
        }
};

int main()
{
    Cube c(2.5f);
    cout<<"Volume of cube is: ";
    c.displayVolume();
    return 0;
}

```

Output -

Volume of cube is: 15.625

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

Program -

```

#include<iostream>
using namespace std;

class Counter
{
    private:
        int c;

```

```

    public:
        Counter()
        {
            c = 0;
        }
        void Count()
        {
            c++;
        }
        void getCount()
        {
            cout<<"Count = "<<c<<endl;
        }
};

int main()
{
    Counter c;
    cout<<"Before calling Count function-\n";
    c.getCount();
    cout<<"After calling Count function-\n";
    c.Count();
    c.getCount();
    return 0;
}

```

Output -

Before calling Count function-
 Count = 0
 After calling Count function-
 Count = 1

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

Program -

```

#include<iostream>
#include<time.h>
using namespace std;

class Date
{
    private:

```

```

        int day, month, year;

public:
    Date()
    {
        day = 1;
        month = 1;
        year = 1900;
    }
    Date(int d, int m, int y)
    {
        day = d;
        month = m;
        year = y;
    }
    void displayDate()
    {
        cout<<"Date: "<<day<<"/"<<month<<"/"<<year<<endl;
    }
};

int main()
{
    Date d(30,9,2000);
    d.displayDate();
    return 0;
}

```

Output -

Date: 30/9/2000

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

Program -

```

#include<iostream>
#include<string.h>
using namespace std;

class Student
{
    private:
        char name[30];

```

```

    int classNum;
    int rollNum;

public:
    Student(char n[], int cn, int r)
    {
        strcpy(name,n);
        classNum = cn;
        rollNum = r;
    }
    void showDetail()
    {
        cout<<"Name: "<<name<<" , Class: "<<classNum<<" , Roll number:
"<<rollNum<<endl;
    }
};

int main()
{
    Student st("Aman Kumar",10,45);
    st.showDetail();
    return 0;
}

```

Output -

Name: Aman kumar, Class: 10, Roll number: 45

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.

Program -

```

#include<iostream>
using namespace std;

class Box
{
    private:
        int length, breadth, height;

    public:

```

```

Box()
{
    length = 0;
    breadth = 0;
    height = 0;
}
void input()
{
    cout<<"Enter length, breadth and height of the Box: ";
    cin>>length>>breadth>>height;
}
void calcVol()
{
    cout<<"Volume of the Box: "<<length*breadth*height<<endl;
}

};

int main()
{
    Box b;
    b.input();
    b.calcVol();
    return 0;
}

```

Output -

Enter length, breadth and height of the Box: 3 6 2
Volume of the Box: 36

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.

Program -

```

#include<iostream>
using namespace std;

class Bank
{
    private:
        float principal;
        float roi;

```



```

        int t;
        float si;

public:
    Bank()
    {
        principal = 0.0f;
        roi = 3.4f;
        t = 0;
        si = 0.0f;
    }
    void input()
    {
        cout<<"Enter principal amount and year: ";
        cin>>principal>>t;
    }
    void calcSI()
    {
        si = (principal * roi * t) / 100;
    }
    void showSI()
    {
        cout<<"Simple Interest: "<<si<<endl;
    }
};

int main()
{
    Bank b;
    b.input();
    b.calcSI();
    b.showSI();
    return 0;
}

```

Output -

Enter principal amount and year: 4534.5 5
Simple Interest: 770.865

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.

Program -

```
#include<iostream>
#include<string.h>
using namespace std;

class Bill
{
    private:
        char name[30];
        long long int phoneNum;
        char address[100];
        int units;
        float bill;

    public:
        Bill()
        {
            strcpy(name,"Not entered");
            phoneNum = 0;
            strcpy(address,"Not entered");
            units = 0;
            bill = 0.0f;
        }
        void get()
        {
            cout<<"Enter your name: ";
            cin.getline(name,30);
            cout<<"Enter your phone number: ";
            cin>>phoneNum;
            cout<<"Enter your address: ";
            cin.ignore();
            cin.getline(address,100);
        }
        void calculateBill()
        {
            cout<<"Enter units: ";
            cin>>units;

            if(units <= 100)
            {
                bill = units * 1.2;
            }
        }
    }
```

```

        else if (bill <= 200)
        {
            bill = (100 * 1.2) + (units - 100) * 2;
        }
        else
        {
            bill = (100 * 1.2) + (100 * 2) + (units - 200) * 3;
        }
    }

    void showBill()
    {
        cout<<"Name of the customer: "<<name<<endl;
        cout<<"Phone number: "<<phoneNum<<endl;
        cout<<"Address: "<<address<<endl;
        cout<<"Units: "<<units<<endl;
        cout<<"Total Bill = Rs. "<<bill;
    }
};

int main()
{
    Bill b;
    b.get();
    b.calculateBill();
    b.showBill();
    return 0;
}

```

Output -

Enter your name: Gopal Singh
 Enter your phone number: 8727281727
 Enter your address: Near XYZ hotel, LKO, UP
 Enter units: 250
 Name of the customer: Gopal Singh
 Phone number: 8727281727
 Address: Near XYZ hotel, LKO, UP
 Units: 250
 Total Bill = Rs. 420

10. Define a class StaticCount and create a static variable. Increment this

Program -

```
#include<iostream>
using namespace std;

class StaticCount
{
    private:
        static int K;

    public:
        static void incrementK()
        {
            K++;
            cout<<"K = "<<K<<endl;
        }
};

int StaticCount::K;

int main()
{
    StaticCount::incrementK();
    StaticCount::incrementK();
    return 0;
}
```

Output -

K = 1

K = 2