

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:
```

```
    Complex()
```

```
    {
```

```
    };
```

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

```
    {
```

```
        a = x;
```

```
        b = y;
```

```
    }
```

```
    void showData()
```

```
    {
```

```
        cout<<"Number = "<<a<<" + "<<b<<"i"<<endl;
```

```
    }
```

```
    Complex add(Complex c)
```

```
    {
```

```
        Complex temp;
```

```
        temp.a = a + c.a;
```

```
        temp.b = b +c.b;
```

```
        return temp;
```

```

    }
};

int main()
{
    Complex c1,c2;
    c1.setData(2,3);
    c2.setData(10,5);
    Complex 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
{
    int HR,MIN,SEC;
    int temp;
public:
    setTime(int h,int m,int s)
    {
        HR = h;
        MIN = m;
        SEC = s;
    }
}

```

```

    }

    Time()

    {

    }

    void showTime()

    {

        cout<<HR<<": "<<MIN<<": "<<SEC<<endl;

    }

    void normalize()

    {

        MIN = MIN + SEC/60;

        SEC = SEC%60;

        HR = HR + MIN/60;

        MIN = MIN%60;

    }

    Time add(Time t)

    {

        Time temp;

        temp.SEC = SEC + t.SEC;

        temp.MIN = MIN + t.MIN;

        temp.HR = HR + t.HR;

        temp.normalize();

        return (temp);

    }

};

int main()

{

    Time t1,t2,t3;

```

```

t1.setTime(4,57,78);
t2.setTime(2,5,54);
t3 = t1.add(t2);
t1.showTime();
t2.showTime();
cout<<"After adding time"<<endl;
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
{
public:
    int side;
    cube(int side1)
    {
        side = side1;
    }
    Volume()
    {
        return(side*side*side);
    }
    cube()
    {
        cout<<"Default constructor is called"<<endl;
    }
};

```

```

int main()
{
    cube c2(5);

    c2.side;

    cout<<"Volume of Cube is : "<<c2.Volume()<<endl;

    return 0;
}

```

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

```

#include<iostream>

using namespace std;

class Counter
{
    static int count;

public:
    Counter()
    {
        count++;

        cout<<"Constructor is called "<<endl;
    }

    int get_count()
    {
        return count++;
    }
};

int Counter::count;

int main()
{
    Counter c1,c2;

    cout<<c1.get_count()<<endl;
}

```

```
    return 0;
}
```

**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 dd,mm,yy;
public:
    Date(int d,int m,int y )
    {
        cout<<"Constructor is called"<<endl;

        dd = d;

        mm = m;

        yy = y;
    }

    void display()
    {
        cout<<"The intered date is "<<dd<<"-"<<mm<<"-"<<yy<<endl;
    }
};

int main()
{
    Date date1(27,1,2023);

    date1.display();

    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
{
    int marks;
    char grade;
public:
    Student(int m, char g)
    {
        marks = m;
        grade = g;
    }
    void display()
    {
        cout<<"marks = "<<marks<<endl;
        cout<<"Grade = "<<grade<<endl;
    }
};

int main()
{
    Student s1(480,'A'),s2(300,'B');
    cout<<"Record of 1st student is : ";
    s1.display();
    cout<<"Record of 2nd student is : ";
    s2.display();
    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
{
    int length,width,height;

    int volume;
public:
    Box(int l,int w,int h)
    {
        length = l;
        width = w;
        height = h;
        volume = length*width*height;
    }

    void vol()
    {
        cout<<"\nDimenions of box are : "<<endl;
        cout<<"Length of box = "<<length<<endl;
        cout<<"Width of box = "<<width<<endl;
        cout<<"Heigth of box is = "<<height<<endl;
        cout<<"Volume of box = "<<volume<<endl;
    }
};

int main()
{
    Box a(5,6,7),b(2,3,4);

    a.vol();
```



```
b.vol();  
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  
{  
    float p,r,t,si,amount;  
public:  
    void read()  
    {  
        cout<<"Enter principle Amount :: ";  
        cin>>p;  
        cout<<"Enter Rate of Interest : ";  
        cin>>r;  
        cout<<"Enter number of Year : ";  
        cin>>t;  
    }  
    void show()  
    {  
        si = (p*r*t)/100;  
        amount = si + p;  
        cout<<"Simple interest = "<<si<<"rupees"<<endl;  
        cout<<"Total Amount = "<<amount<<"rupees"<<endl;  
        cout<<"\n Thank you "<<endl;  
    }  
}
```

```
};

int main()
{
    bank x;

    x.read();

    x.show();

    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.**

```
#include<iostream>

using namespace std;

class electricityBill
{
    int c_no;

    char c_name[20];

    int units;

    int bill;

public:

    void put()
    {
        cout<<"Enter Details of customer ::"<<endl;

        cout<<"\nEnter customer number :";

        cin>>c_no;

        cout<<"Enter name of customer :";

        cin>>c_name;
```

```

        cout<<"Enter number of unit used :";

        cin>>units;
    }
    void get()
    {
        cout<<"\nEntered Details of customer is : "<<endl;

        cout<<"Customer no. is = "<<c_no<<endl;

        cout<<"Customer name is = "<<c_name<<endl;

        cout<<"Number of Units used by Customer : "<<units<<endl;

        cout<<"Bill of customer is = "<<bill<<endl;
    } void calculate_bill()
    {
        if(units<=100)
            bill = units*1.20;
        else if(units<=200)
            bill = 100*1.20 +(units-100)*2;
        else
            bill = 100*1.20 + 100*2 +(units-200)*3;
    }
};

int main()
{
    electricityBill x;

    x.put();

    x.calculate_bill();

    x.get();

    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
{
    static int count;
    int x;
public:
    Staticcount()
    {
        cout<<"Constructor is called "<<endl;
        count++;
    }
    void increment()
    {
        x = count++;
    }
    void display()
    {
        cout<<"Number of calling function is "<<count<<endl;
    }
};

int Staticcount::count;

int main()
{
    Staticcount c1;
    c1.increment();
    c1.display();
```

```
return 0;
```

```
}
```