अध्याय 9

क्लास और ऑब्जेक्ट

9.1 परिचय

क्लास ऑब्जेक्ट ऑरिएनटेड प्रोग्रामिंग भाषा का एक महत्वपूर्ण फीचर है। क्लास की अवधारणा को C में स्ट्रक्चर से लिया गया है। यह एक यूजर डिफान्ड टाईप को बनाने का और लागू करने का नया तरीका है। इस अध्याय में हम क्लास और ऑब्जेक्ट के विभिन्न अवधारणाओं पर चर्चा करेगें।

9.2 क्लास को परिभाषित करना

क्लास एक यूजर डिफान्ड टाईप है जो डेटा और फंक्शन को एक साथ बाँधे रखता है। क्लास की घोषणा में इसके डेटा और मेम्बर फंक्शन की घोषणा होती है। क्लास की घोषणा का प्रारूप इस प्रकार होता है –

```
class class_name
{
  private:
        variable declaration;
        function declaration;
public:
         variable declaration;
        function declaration;
};
```

क्लास के मेम्बर जिसकी घोषणा प्राईवेट अनुभाग में है उनको उसी क्लास के मेम्बर ही एक्सेस कर सकते है। क्लास के मेम्बर जिनकी घोषणा पब्लिकअनुभाग में हुई है उनको क्लास के बाहर से एक्सेस कर सकते है। स्वतः ही क्लास के मेम्बर प्राईवेट होते है। डेटा को क्लास के प्राईवेट अनुभाग में घोषित करना डेटा हाइडिंग कहा जाताहै औरयह ऑब्जेक्ट ऑरिएनटेड प्रोग्रामिंग भाषा का एक महत्वपूर्ण फिचर है।

क्लास का एक उदाहरण

```
class point
{
        int x,y;                // private by default
```

```
public:
        void input(int a, int b);
        void output(void);
};
```

ऑब्जेक्ट की घोषणा

बेसिक डेटा टाईप की तरह हम क्लास टाईप के वेरिएबल की घोषणाकर सकते है। इन वेरिएबल को ऑब्जेक्ट कहा जाता है।

उदाहरण के लिए :–

```
point p, q;
```

यहाँ दो ऑब्जेट p और qघोषित कियेगये है।

क्लास के मेम्बर को एक्सेस करना

हम केवल क्लास के पब्लिक मेम्बरस को उस क्लास के ऑब्जेक्ट के द्वारा एक्सेस कर सकते है। पब्लिक मेंम्बर फंक्शन को एक्सेस करने का प्रारूप इस प्रकार है

```
object_name.function_name(arguments list);
```

उदाहरण के लिए

```
p.input(10,20);
```

ऑब्जेक्ट pके द्वारा input()फंक्शन को कॉलकिया गया है। स्टेटमेंट p.x=10; मान्य नहीं है क्योंकि x को प्राईवेट घोषित किया गया है और इसे केवल क्लास के मेंम्बर फंक्शन ही सीधे एक्सेस कर सकते है न की ऑब्जेक्ट के द्वारा ।

9.3 मेम्बर फंक्शन को परिभाषित करना

क्लास के मेंम्बर फंक्शन को क्लास के अन्दर और क्लास से बाहर परिभाषित किया जा सकता है।

क्लास के अन्दर

मेंम्बर फंक्शन की घोषणा को क्लास के अन्दर उसकी वास्तविक परिभाषा से विस्थापित किया जाता है। क्लास के अन्दर परिभाषित मेंम्बर फंक्शन इनलाईन फंक्शन माने जाते है।
उदाहरण के लिए :–

```
class point
{
        int x,y;
public:
        void input(int a, int b)
```

```
        {
                x=a;
                y=b;
        }
        void output(void)
        {
                cout<<"x="<<x<<"\n";
                cout<<"y="<<y;
        }
};
```

क्लास के बाहर

मेंम्बर फंक्शन जिनकी घोषणा क्लास के अन्दर की गयी हो उनको क्लास से बाहर अलग से परिभाषित करना होता है।

मेम्बर फंक्शन को परिभाषित करने के लिए प्रारूप :–

```
return_type class_name:: function_name(arguments)
{
        function body
}
```

यहाँ class_name दर्शाता है कि फंक्शन इस क्लास से सम्बंधित है।

उदाहरण के लिए :–

```
class point
{
        int x,y;
public:
        void input(int a, int b);
        void output(void);
};
void point :: input(int a, int b)
        {
                x=a;
                y=b;
        }
```

```
        void point :: output(void)
{
        cout<<“x=”<<x<<“\n”;
        cout<<“y=”<<y;
}
```

प्रोग्राम 9.1 क्लास के साथ एक प्रोग्राम :–

```
#include<iostream>
using namespace std;
class point
{
        int x,y;
public:
        void input(int a, int b);
        void output(void);
};
void point :: input(int a, int b)
        {
                x=a;
                y=b;
        }
        void point :: output(void)
        {
                cout<<“x=”<<x<<“\n”;
                cout<<“y=”<<y;
        }
int main()
{
        point p;
        p.input(5,10);
        p.output();
return 0;
}
```

प्रोग्राम 9.1 का आउटपुट होगा–

x=5

y=10

9.4 एक्सेस मोडिफायरस

public और private कीवर्डस को एक्सेस मोडिफायर्स कहा जाता है। चूकि ये क्लास के मेंम्बर को एक्सेस करने की प्रणाली को नियंत्रित करते है। क्लास के पब्लिक मेंम्बर को क्लास के बाहर से एक्सेस किया जा सकता हैं सामान्यतः क्लास के मेम्बर फंक्शन को पब्लिक अनुभाग में रखा जाता है। क्लास के प्राईवेट मेंम्बर क्लास के बाहर से एक्सेस नहीं किये जा सकतें है। यहाँ तक कि उस क्लास के ऑब्जेक्ट के द्वारा भी नहीं किये जा सकते हैं। सामान्यतः क्लास के डेटा मेंम्बर को प्राईवेट अनुभाग में रखा जाता हैं।

9.5 क्लास के भीतर ऐरे

ऐरे भी क्लास के डेटा मेंम्बर के रूप में हो सकते है।

उदाहरण के लिए :–

प्रोग्राम 9.2 क्लास के भीतर ऐरे

```
#include<iostream>
using namespace std;
class data
{
        int a[5];
public:
        void getdata(void);
        void showdata(void);
};
void data :: getdata(void)
{
        cout<<"Enter the elements of array\n";
        for(int i=0; i<5; i++)
        {
          cin>>a[i];
        }
}
void data :: showdata(void)
```

```
{
        cout<<"Array elements are\n";
        for(int i=0; i<5; i++)
        cout<<a[i]<<"\t";
}
int main()
{
    data d;
    d.getdata();
    d.showdata();
return 0;
}
```

प्रोग्राम 9.2 का आउटपुट होगा–

```
Enter the elements of array
6       5       9       8       1
Array elements are
6       5       9       8       1
```

9.6 स्टेटिक डेटा मेंम्बर

क्लास के डेटा मेंम्बर को स्टेटिक के रूप में भी घोषित किया जा सकता है। स्टेटिक डेटा मेंम्बर की विशेषताएँ इस प्रकार है –

- इसकी प्रांरभिक वेल्यू शून्य होती है जब इसके क्लास का पहला ऑब्जेक्ट बनाया जाता है।
- केवल एक ही प्रतिलिपी इस डेटा मेंम्बर की बनती है और इसे क्लास के सभी ऑब्जेक्ट साझा करते है।
- चूंकि यह सम्पूर्ण क्लास के साथ जुड़ा हुआ है इसे क्लास वेरिएबल भी कहा जाता है।

प्रोग्राम 9.3 स्टेटिक डेटा मेंम्बर

```
#include<iostream>
using namespace std;
class data
{
        static int x;
```

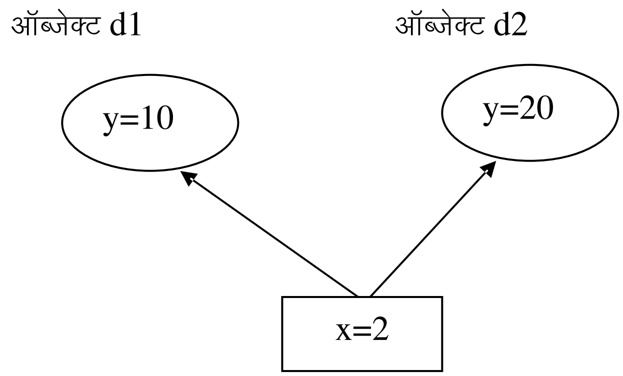
```
    int y;
    public:
    void getdata(int a)
    {
        y=a;
        x++;
    }
    void show_x(void)
    {
        cout<<“x=”<<x<<“\n”;
    }
};
int data :: x;   // static member definition
int main()
{
    data d1, d2;   // x is initialized to zero
    d1.show_x();
    d2.show_x();
    d1.getdata(10);
    d2.getdata(20);
    cout<<“After reading data”<<“\n”;
    d1.show_x();
    d2.show_x();
    return 0;
}
```

प्रोग्राम 9.3 का आउटपुट होगा–

```
x=0
x=0
After reading data
x=2
x=2
```

उपरोक्त प्रोग्राम में जब ऑब्जेक्ट घोषित किये जाते है तब स्टेटिक डेटा मेंम्बर x की प्रारंभिक वेल्यू शून्य हो जाती है। हर बार जब फंक्शन कॉल होता है तब x की वेल्यू में एक की वृद्वि हो जाती है। चूंकि वेरिएबल x को दोनों ऑब्जेक्ट के बीच साझा किया गया है। x का मान हर बार 2 प्रिन्ट हुआ है। स्टेटिक डेटा मेंम्बर को प्रारंभिक वेल्यू भी दी जा सकती है जब इसे क्लास के बाहर परिभाषित किया जाता है। जैसे

```
int data :: x=5;
```



चित्र 9.1 स्टेटिक डेटा मेंम्बर

9.7 स्टेटिक मेंम्बर फंक्शन

स्टेटिक कीवर्ड के साथ घोषित मेंम्बर फंक्शन को स्टेटिक मेंम्बर फंक्शन कहा जाता है। स्टेटिक मेंम्बर फंक्शन की निम्नलिखित विशेषताएँ होती है

- यह केवल क्लास के दूसरे स्टेटिक डेटा मेंम्बर और मेंम्बर फंक्शन को एक्सेस कर सकते है।
- इनको क्लास के नाम की सहायता से कॉल किया जाता है।

प्रोग्राम 9.4 स्टेटिक मेंम्बर फंक्शन

```
#include<iostream>
using namespace std;
class test
{
	int x;
	static int y;
public:
	void set_xy(int a)
	{
		x=a;
```

```
		y++;
	}
	void show_x(void)
	{
		cout<<"x="<<x<<"\n";
	}
	staic void show_y(void)
	{
		cout<<"y="<<y;
	}
};
int test :: y;
int main()
{
test t1, t2;
t1.set_xy(10);
t2.set_xy(20);
t1.show_x();
t2.show_x();
test::show_y();		// calling static function
return 0;
}
```

प्रोग्राम 9.4 का आउटपुट होगा–

```
x=10
x=20
y=2
```

9.8 फ्रेंड फंक्शन

जैसा की हम जानते है कि क्लास के प्राईवेट मेंम्बर को क्लास के बाहर से एक्सेस नहीं किये जा सकते है। एक फ्रेंड फंक्शन क्लास के प्राईवेट डाटा को उस क्लास के ऑब्जेक्ट के जरिए एक्सेस कर सकते हैं। जब एक फंक्शन दो क्लासों में एक समान हो, सामान्यतः हम उस फंक्शन को दोनों के लिए फ्रेंड बना लेते है। इस फंक्शन को friend कीवर्ड के साथ घोषित किया जाता है। एक फ्रेंड फंक्शन की निम्नलिखित विशेषताएँ होती है

- इसे सामान्य फंक्शन की तरह कॉल किया जाता है।

- क्लास के ऑब्जेक्ट की सहायता से कॉल नहीं किया जा सकता हैं।
- यह क्लास के मेंम्बर को केवल उस क्लास के ऑब्जेक्ट की सहायता से एक्सेस कर सकता है।
- इसे क्लास के अन्दर कही भी घोषित किया जा सकता है।
- सामान्यतः इसके आरग्यूमेन्ट ऑब्जेक्ट होते है।

प्रोग्राम 9.5 फ्रेंड फंक्शन

```
#include<iostream>
using namespace std;
class test
{
        int  x,y;
        public:
        void getdata(int a, int b)
        {
                x=a;
                y=b;
        }
        friend int sum(test t);
};
        int sum(test t)
        {
                return(t.x+t.y);
        }
        int main()
        {
                test q;
                q.getdata(10,20);
                cout<<"Sum="<<sum(q);
                return 0;
        }
```

प्रोग्राम 9.5 का आउटपुट होगा–

Sum=30

फ्रेंड क्लास

एक क्लास का मेंम्बर फंक्शन दूसरी क्लास का फ्रेंड फंक्शन हो सकता है।

उदाहरण के लिए :–

```
class A
{
void fun();     // member function of A
};
class B
{
- ------
friend void A::fun();
--------
};
```

ध्यान दे कि क्लास B में फ्रेंड फंक्शन को क्लास का नाम और स्कोप रिजोलुयशन ऑपरेटर के साथ घोषित किया गया है। फंक्शन fun() क्लास A का मेंम्बर फंक्शन है और क्लास B का फ्रेंड फंक्शन है। अगर एक क्लास के सभी मेंम्बर फंक्शन दूसरी क्लास में फ्रेंड घोषित कर दिये जाते हैं तब उस क्लास को फ्रेंड क्लास कहा जाता है।

उदाहरण के लिए :–

```
class C
{
- ------
friend class A;                //All member functions of class A are friend
to C
- ------
};
```

प्रोग्राम 9.6 फ्रेंड फंक्शन

```
#include<iostream>
using namespace std;
class second; //forward declaration
class first
{
        int x;
public:
```

```
        void set_value(int a)
        {
                x=a;
        }
        friend void max(first, second);
};
class second
{
        int y;
public:
        void set_value(int b)
        {
                y=b;
        }
        friend void max(first, second);
};
void max(first f, second s)
{
        if(f.x>s.y)
        cout<<"Maximum is"<<f.x;
        else
        cout<<"Maximum is"<<s.y;
}
int main()
{
        first A;
        second B;
        A.set_value(10);
        B.set_value(20);
        max(A,B);     // calling friend function
        return 0;
}
```

प्रोग्राम 9.6 का आउटपुट होगा–

Maximum is 20

9.9 रिटर्निगं ऑब्जेक्ट

पिछले अनुभाग में हमने देखा कि फ्रेंड फंक्शन ऑरग्यूमेन्ट के रूप में ऑब्जेक्ट लेते हैं। फ्रेंड फंक्शनऑब्जेक्ट भी रिटर्न कर सकते हैं।

प्रोग्राम 9.7 रिटर्निगं ऑब्जेक्ट

```
#include<iostream>
using namespace std;
class vector
{
	int V[3];
public:
	void set_vector(void)
	{
		cout<<"Enter three numbers\n";
		for(int i=0; i<3; i++)
		cin>>V[i];
	}
	void display(void)
	{
		for(int i=0; i<3; i++)
		cout<<V[i]<< ",";
	}
	friend vector sum(vector, vector);
};
vector sum(vector p, vector q)
{
	vector r;
	for(int j=0;j<3; j++)
	r.V[j]=p.V[j]+q.V[j];
	return r;
}
int main()
```

```
{
        vector v1, v2,v3;
        v1.set_vector();
        v2.set_vector();
        v3=sum(v1,v2);
        cout<<"First vector is:";
        v1.display();
        cout<<"\n";
        cout<<"Second vector is:";
        v2.display();
        cout<<"\n";
        cout<<"Resultant vector is:";
        v3.display();
        return 0;
}
```

प्रोग्राम 9.7 का आउटपुट होगा–

```
Enter three numbers
3       -2      5
Enter three numbers
-8      6       7
First vector is: 3,-2,5,
Second vector is: -8,6,7,
Resultant vector is: -5,4,12
```

9.10 मेंम्बर के लिए पोइन्टर

हम एक क्लास के मेंम्बर का एड्रेस पोइंटर को असाइन कर सकते है।

उदाहरण के लिए :–

```
class X
{
        int a;
public:
        void show();
};
```

हम क्लास X के मेंम्बर a के लिए पोइंटर परिभाषित इस प्रकार कर सकते हैं–

```
int X:: *p=&X::a;
```

X:: * का मतलब क्लास X के मेंम्बर के लिए पोंइटर।

&X:: * का मतलब क्लास X का मेंम्बर a का एड्रेस।

स्टेटमेंट int *p=&a; कार्य नहीं करेगा। पोइंटर p का उपयोग डेटा मेंम्बर a को मेंम्बर फंक्शन और फ्रेंड फंक्शन के अन्दर एक्सेस करने के लिए कर सकते है।

उदाहरण के लिए :–

```
void show()
{
        X x;   // object created
        cout<<x.*p;  //display value of a
        cout<<x.a;   //same as above
}
```

हम क्लास के मेंम्बर फंक्शन के लिए पोंइटर सेट कर सकते है। मेंम्बर फंक्शन को डिरेफरेंसिंग ऑपरेटर (.*) की सहायता से कॉल कर सकते हैं।

उदाहरण के लिए :–

```
X x;            // object created
void (X::*pf)()=&X::show;
(x.*pf)();      //invoke show()
```

यहाँ pf मेंम्बर फंक्शन show() के लिए पोइंटर हैं।

महत्वपूर्ण बिंदु

- क्लास एक यूजर डिफान्ड टाईप है जो डेटा और फंक्शन को एक साथ बाँधे रखता है।
- स्वतः ही क्लास के मेम्बर प्राईवेट होते है।
- क्लास के मेंम्बर फंक्शन को क्लास के अन्दर और क्लास से बाहर परिभाषित किया जा सकता है।
- public और private कीवर्ड को एक्सेस मोडिफायर्स कहा जाता है।
- क्लास के डेटा मेंम्बर को स्टेटिक के रूप में भी घोषित किया जा सकता है।
- स्टेटिक कीवर्ड के साथ घोषित मेंम्बर फंक्शन को स्टेटिक मेंम्बर फंक्शन कहा जाता है।

- एक फ्रेंड फंक्शन क्लास के प्राईवेट डाटा को उस क्लास के ऑब्जेक्ट के जरिए एक्सेस कर सकते हैं।
- हम एक क्लास के मेंम्बर का एड्रेस पोइंटर को असाइन कर सकते है।

अभ्यासार्थ प्रश्न

वस्तुनिष्ठ प्रश्न :

प्रश्न 1. एक यूजर डिफान्ड टाईप जो डेटा और फंक्शन को एक साथ बाँधे रखता है उसेकहा जाता है

(अ) ऑब्जेट (ब) क्लास (स) ऐरे (द) पोइंटर

प्रश्न 2. स्वतः ही क्लास के मेम्बर होते है

(अ) पब्लिक (ब) प्राईवेट (स) प्रोटेक्टेड (द) इनमें से कोई नहीं

प्रश्न 3. इनमें से कोनसा एकएक्सेस मोडिफायर है?

(अ) public (ब) private (स) अ. और ब. दोनों (द) इनमें से कोई नहीं

प्रश्न 4. इनमें से कोनसा स्टेटिक डेटा मेंम्बर के संदर्भ में सत्य है ?

(अ) इसकी प्रारंभिक वेल्यू शून्य होती है जब इसके क्लास का पहला ऑब्जेक्ट बनाया जाता है।

(ब) केवल एक ही प्रतिलिपी इस डेटा मेम्बर की बनती है।

(स) इसे क्लास वेरिएबल भी कहा जाता है।

(द) उपरोक्त सभी

प्रश्न 5 इनमें से कोनसा स्टेटिक मेंम्बर फंक्शन के संदर्भ में सत्य है?

(अ) स्टेटिक कीवर्ड के साथ घोषित किया जाता है।

(ब) केवल क्लास के दूसरे स्टेटिक डेटा मेंम्बर और मेंम्बर फंक्शन को एक्सेस कर सकते है।

(स) इनको क्लास के नाम की सहायता से कॉल किया जाता है।

(द) उपरोक्त सभी

प्रश्न 6. इनमें से कोनसा फ्रेंड फंक्शन के संदर्भ में सत्य है?

(अ) इसे सामान्य फंक्शन की तरह कॉल किया जाता है।

(ब) इसे क्लास के अन्दर कही भी घोषित किया जा सकता है।

(स) सामान्यतः इसके आरग्यूमेन्ट ऑब्जेक्ट होते है।

(द) उपरोक्त सभी

अति लघूउत्तरात्मक प्रश्न

प्रश्न 1. क्लास किसे कहते है ?

प्रश्न 2. ऑब्जेक्ट किसे कहते है ?

प्रश्न 3. फ्रेंड क्लास किसे कहते है ?

लघूउत्तरात्मक प्रश्न

प्रश्न 1. प्राईवेट और पब्लिक एक्सेस मोडिफायर में क्या अन्तर है ?

प्रश्न 2. स्टेटिक डेटा मेम्बर के गुण क्या है ?

प्रश्न 3. स्टेटिक मेंम्बर फंक्शन के क्या गुण होते है ?

निबंधात्मक प्रश्न

प्रश्न 1. फ्रेंड फंक्शन किसे कहते है ? इसके गुण लिखों ?

प्रश्न 2. एक 'Complex' क्लास बनाइए जो एक Complex नम्बर को बताता है और दो कॉम्पलेक्स नम्बरों को जोड़ने और घटाने के लिए मेंम्बर फंक्शन परिभाषित करने का प्रोग्राम लिखो ?

प्रश्न 3. दो क्लासों के डेटा मेंम्बर की अदला–बदली करने का फ्रेंड फंक्शन की सहायता से प्रोग्राम लिखों ?

उत्तरमाला

1:ब 2:ब 3:स 4: द 5: द 6:द