अध्याय –11

ऑपरेटर ओवरलोडिंग

11.1 परिचय

ऑपरेटर ओवरलोडिंग C++ भाषा का महत्पवपूर्ण फिचर है। इस फिचर से हम यूजर–डिफान्ड टाईप के दो वेरिएबल को जोड़ सकते है उसी तरह से जिस तरह से हम बेसिक डेटा टाईप से करते है। किसी ऑपरेटर को एक डाटा टाईप के लिए विशेष मिनिंग देना ऑपरेटर ओवरलोडिंग कहा जाता है। C++ में सभी ऑपरेटर को ओवरलोड कर सकते है। सिवाय निम्नलिखित ऑपरेटर के –

- क्लास मेंम्बर एक्सेस ऑपरेटर (., .*)
- स्कोप रिजोल्यूशन ऑपरेटर (::)
- साइज ऑपरेटर (sizeof)
- कंडिशनल ऑपरेटर (?:)

जब हम एक ऑपरेटर को ओवरलोड करते है इसका मूलरूप बरकरार रहता है। उदाहरण के लिए अगर हम + ऑपरेटर को दो मैट्रीक्स को जोडने के लिए करते है तब भी इस ऑपरेटर से दो संख्याओं को भी जोड़ सकते है।

11.2 ऑपरेटर फंक्शन

एक ऑपरेटर को अतिरिक्त मिनिंग देने के लिए हम एक विशेष फंक्शन काम में लेते है जिसे ऑपरेटर फंक्शन कहा जाता है। ऑपरेटर फंक्शन का प्रोटोटाईप इस प्रकार होता हैं–

```
return_type class_name :: operator op(arguments list)
{
        function body
}
```

जहाँ operator एक कीवर्ड है और op एक ऑपरेटर है जिसे ओवरलोड किया जाना है। ऑपरेटर फंक्शन एक क्लास का मेंम्बर फंक्शन या फ्रेंड फंक्शन होना चाहिए। उनमें फर्क यह है कि मेंम्बर फंक्शन यूनरी ऑपरेटर के लिए कोई आरग्यूमेंन्ट नहीं लेता है और बाइनरी ऑपरेटर के लिए एक आरग्यूमेन्ट लेता है जबकि फ्रेंड फंक्शन यूनरी ऑपरेटर के लिए एक आरग्यूमेन्ट लेता है और बाइनरी ऑपरेटर के लिए दो ऑपरेटर लेता है।

11.3 यूनरी ऑपरेटर को मेंम्बर फंक्शन से ओवरलोड करना

हम पोस्ट फिक्स इंक्रिमेंट ऑपरेटर (++) को उदाहरण के लिए लेते है। यह केवल एक ऑपरेड लेता हैं और इसकी वेल्यू को एक से बढ़ा देता है जब बेसिक डेटा टाईप के साथ प्रयोग किया जाता है। हम इस ऑपरेटर को ओवरलोड करेंगें ताकि इसका प्रयोग एक ऑब्जेक्ट के साथ किया जा सके और इस ऑब्जेक्ट के हर एक डेटा आइटम की वेल्यू को एक बढ़ा देगा।

प्रोग्राम11.1 पोस्ट फिक्स इंक्रिमेंट ऑपरेटर को ओवरलोड करना

```
#include<iostream>
using namespace std;
class point
{
        int x,y;
public:
         void getdata(int a, int b)
         {
                  x=a;
                  y=b;
         }
        void show(void)
         {
               cout<<"x="<<x;
               cout<<"y="<<y<<"\n";
         }
        void operator++(int)
        {
               x++;
               y++;
        }
};

int main()
{
  point p;
  p.getdata(5,8);
```

```
  cout<<"p:";
  p.show();
  p++;          // invoke operator function
  cout<<"p++:";
  p.show();
return 0;
}
```

प्रोग्राम 11.1 का आउटपुट होगा–

```
p: x=5 y=8
p++: x=6 y=9
```

ऑपरेटर फंक्शन में int का प्रयोग यह दर्शाता है कि हम पोस्टफिक्स इंक्रिमेंट ऑपरेटर को ओवरलोड कर रहे है न कि प्रिफिक्स इंक्रिमेंट ऑपरेटर को।

11.4 यूनरी ऑपरेटर को फ्रेंड फंक्शन से ओवरलोड करना

हम प्रिफिक्स डिक्रिमेंट ऑपरेटर (--) को उदाहरण के लिए लेते हैं। जो एक ऑपरेड लेता है और उसकी वेल्यू को एक कम कर देता है जब बेसिक डेटा टाईप के साथ प्रयोग किया जाता हैं। हम इस ऑपरेटर को ओवरलोड करेंगें ताकि इसका प्रयोग ऑब्जेक्ट के साथ किया जा सके और इस ऑब्जेक्ट के हर एक डेटा आइटम की वेल्यू को एक कम कर देगा।

प्रोग्राम 11.2 :– प्रिफिक्स डिक्रिमेंट ऑपरेटर को ओवरलोड करना

```
#include<iostream>
using namespace std;
class point
{
        int x,y;
public:
          void getdata(int a, int b)
          {
                  x=a;
                  y=b;
          }
         void show(void)
          {
                 cout<<"x="<<x;
```

```
            cout<<"y="<<y<<"\n";
        }
        friend void operator--(point &s)
        {
            s.x=s.x-1;
            s.y=s.y-1;
        }
};
int main()
{
  point p;
  p.getdata(7,10);
 cout<<"p:";
  p.show();
  --p;
 cout<<"--p:";
  p.show();
return 0;
}
```

प्रोग्राम 11.2 का आउटपुट होगा–

p: x=7 y=10

p++: x=6 y=9

ध्यान दे कि ऑपरेटर फंक्शन में आरग्यूमेन्ट रेफरेंस के द्वारा भेजा गया है। अगर हम वेल्यू के द्वारा भेजते यह काम नहीं करता क्योंकि जो बदलाव ऑपरेटर फंक्शन में किये गये main() फंक्शन में प्रतिबिम्बित नहीं होगें। निम्नलिखित ऑपरेटर्स को फ्रेंड फंक्शन से ओवरलोड नहीं किया जा सकता है–

- असाइनमेंट ऑपरेटर =
- फंक्शन कॉल ऑपरेटर ()
- सब्सक्रिप्टींग ऑपरेटर []
- क्लास मेंम्बर एक्सेस ऑपरेटर ->

11.5 बाइनरी ऑपरेटर को मेंम्बर फंक्शन से ओवरलोड करना

हम बाइनरी प्लस ऑपरेटर (+) को उदाहरण के लिए लेते है। जो दो ऑपरेंड लेता है और दोनों का योग कर देता है। जब इसका प्रयोग बेसिक डेटा टाईप के साथ किया जाता है। हम इस ऑपरेटर को दो मैट्रिक्स को जोड़ने के लिए उपयोग करेंगे।

प्रोग्राम 11.3 :– बाइनरी प्लस ऑपरेटर (+) को ओवरलोड करना

```
#include<iostream>
using namespace std;
class matrix
{
        int mat[2][2];
public:
          void getmatrix(void);
          matrix operator+(matrix);
          void showmatrix(void);
};
void matrix::getmatrix(void)
 {
   for(int i=0; i<2; i++)
     for(int j=0; j<2; j++)
       {
         cout<<"Enter the number:";
         cin>>mat[i][j];
         }
 }
 matrix matrix::operator+(matrix m)
 {
  matrix temp;
  for(int i=0; i<2; i++)
     for(int j=0; j<2; j++)
       temp.mat[i][j]=mat[i][j]+m.mat[i][j];
  return temp;
 }
```

```
 void matrix::showmatrix(void)
 {
   for(int i=0; i<2; i++)
     {
       for(int j=0; j<2; j++)
       cout<<mat[i][j]<<“\t”;
       cout<<“\n”;
     }
}

int main()
{
  matrix m1,m2,m3;
  m1.getmatrix();
  m2.getmatrix();
  m3=m1+m2;
  cout<<"matrix m1:\n";
  m1.showmatrix();
  cout<<"matrix m2:\n";
  m2.showmatrix();
  cout<<"Resultant matrix:\n";
   m3.showmatrix();
return 0;
}
```

प्रोग्राम 11.3 का आउटपुट होगा–

```
Enetr the number: 2
Enetr the number: 3
Enetr the number: 1
Enetr the number: 4
Enetr the number: 6
Enetr the number: 7
Enetr the number: 8
```

```
Enetr the number: 9
matrix m1:
2       3
1       4
matrix m2:
6       7
8       9
Resultant matrix:
8       10
9       13
```

उपरोक्त प्रोग्राम में, ऑपरेटर फंक्शन मैट्रिक्स टाईप का एक आरग्यूमेन्ट लेता है। जो कि बाइनरी प्लस ऑपरेटर का दूसरा ऑपरेड है। पहला ऑपरेड m1 का प्रयोग ऑपरेटर फंक्शन को कॉल करने के लिए किया गया है ताकि m1 के डेटा मेंम्बर को ऑपरेटर फंक्शन द्वारा सीधे ही एक्सेस किया गया है।

```
m3=m1+m2;
```

निम्न स्टेटमेंट के समान है

```
m3 =m1.operator+(m2);
```

बाइनरी ऑपरेटर के लिए, बायी तरफ का ऑपरेड ऑपरेटर फंक्शन को कॉल करने के लिए किया जाता है और दायीं तरफ के ऑपरेटर फंक्शन का आरग्यूमेन्ट के रूप में भेजा जाता है।

11.6 बाइनरी ऑपरेटर को फ्रेंड फंक्शन से ओवरलोड करना

निम्नलिखित प्रोग्राम बाइनरी प्लस ऑपरेटर (+) को फ्रेंड फंक्शन से दो कॉम्पलेक्स संख्याओं को जोडनें का उदाहरण है।

प्रोग्राम 11.4 बाइनरी प्लस ऑपरेटर को फ्रेंड फंक्शन से ओवरलोड करना

```
#include<iostream>
using namespace std;
class complex
{
    float real;
    float imag;
pubic:
        void input(float x, float y)
```

```
{
        real=x;
        imag=y;
}
friend complex operator + (complex a, complex b)
{
        complex c;
        c.real=a.real+b.real;
        c.imag=a.imag+b.imag;
        return c;
}
void show(void)
{
        cout<<real<<"+i"<<imag<<"\n";
}
};
int main()
{
        complex c1,c2,c3;
        c1.input(1.6,6.2);
        c2.input(2.3,3.4);
        c3=c1+c2;           //invoke operator function
        cout<<"C1=";
        c1.show();
        cout<<"C2=";
        c2.show();
        cout<<"C3=";
        c3.show();
        return 0;
}
```

प्रोग्राम 11.3 का आउटपुट होगा–

C1=1.6+i6.2;

C2=2.3+i3.4;

C3=3.9+i9.6;

उपरोक्त प्रोग्राम में ऑपरेटर फंक्शन कॉम्पलेक्स टाईप के दो आरग्यूमेन्ट लेता है और एक कॉम्पलेक्स संख्या को परिणाम के रूप में रिटर्न करता है। निम्नलिखित स्टेटमेंट

c3=c1+c2;

निम्न स्टेटमेंट के समान है

c3=operator+(c1,c2);

महत्वपूर्ण बिंदु

- किसी ऑपरेटर को एक डाटा टाईप के लिए विशेष मिनिंग देना ऑपरेटर ओवरलोडिंग कहा जाता है।
- जब हम एक ऑपरेटर को ओवरलोड करते है इसका मूलरूप बरकरार रहता है।
- ऑपरेटर को अतिरिक्त मिनिंग देने के लिए हम एक विशेष फंक्शन काम में लेते है जिसे ऑपरेटर फंक्शन कहा जाता है।
- ऑपरेटर फंक्शन एक क्लास का मेंम्बर फंक्शन या फ्रेंड फंक्शन होना चाहिए।

अभ्यासार्थ प्रश्न

वस्तुनिष्ठ प्रश्न :

प्रश्न 1. किस ऑपरेटर को ओवरलोड नहीं किया जा सकता है?

(अ) स्कोप रिजोल्यूशन ऑपरेटर (::)(ब) क्लास मेंम्बर एक्सेस ऑपरेटर (., .*)
(स) बाइनरी प्लस ऑपरेटर (+) (द) कंडिशनल ऑपरेटर (?:)

प्रश्न 2. बाइनरी ऑपरेटर को ओवरलोड करने के लिए ऑपरेटर फंक्शन मेंम्बर फंक्शन के रूप में कितने आरग्यूमेन्ट लेता है?

(अ) दो आरग्यूमेन्ट (ब) एक आरग्यूमेन्ट
(स) शून्य आरग्यूमेन्ट (द) इनमें से कोई नहीं

प्रश्न 3. यूनरी ऑपरेटर को ओवरलोड करने के लिए ऑपरेटर फंक्शन फ्रेंड फंक्शन के रूप में कितने आरग्यूमेन्ट लेता है?

(अ) दो आरग्यूमेन्ट (ब) एक आरग्यूमेन्ट
(स) शून्य आरग्यूमेन्ट (द) इनमें से कोई नहीं

प्रश्न 4. किस ऑपरेटर को फ्रेंड फंक्शन से ओवरलोड नहीं कर सकते है?

(अ) = असाइनमेंट ऑपरेटर (ब) () फंक्शन कॉल ऑपरेटर
(स)[] सब्सक्रिप्टींग ऑपरेटर (द) उपरोक्त सभी

अति लघूउत्तरात्मक प्रश्न

प्रश्न 1. ऑपरेटर ओवरलोडिंग किसे कहा जाता हैं ?

प्रश्न 2 ऑपरेटर फंक्शन का प्रोटोटाईप लिखों ?

प्रश्न 3. ऑपरेटरस का नाम लिखों जिनको ओवरलोड़ नहीं किया जा सकता है ?

लघूउत्तरात्मक प्रश्न

प्रश्न 1. ऑपरेटर फंक्शन मेंम्बर फंक्शन के रूप में और ऑपरेटर फंक्शन फ्रेंड फंक्शन के रूप में दोनों में अन्तर का वर्णन कीजिए ?

निबंधात्मक प्रश्न

प्रश्न 1. क्लास के ऑब्जेक्ट को त्रृणात्मक बनाने के लिए यूनरी माइनस ऑपरेटर को ओवरलोड करने का प्रोग्राम फ्रेंड फंक्शन का उपयोग करके लिखों ?

प्रश्न 2 बाइनरी प्लस ऑपरेटर को दो स्ट्रींग को जोड़ने के लिए ओवरलोड करने का प्रोग्राम मेंम्बर फंक्शन का उपयोग करके लिखों ?

उत्तरमाला

1:स 2:ब 3:ब 4: द