TATA ELXSI

OBJECT ORIENTED PROGRAMMING USING C++
Module 7

Learning & Development Team

Operator Overloading

Operator Overloading (Contd…)

- Operator overloading is an important concept in C++.
- It is a type of polymorphism in which an operator is overloaded to give user defined meaning to it.
- Overloaded operator is used to perform operation on user-defined data type.
- For example '+' operator can be overloaded to perform addition on various data types, like for Integer, String(concatenation) etc.

Operator Overloading

- Almost any operator can be overloaded in C++.

- These below operators can be overloaded.
  - new delete
  - + - * / % ^ & | ~
  - ! = < > += -= *= /= %=
  - ^= &= |= << >> >>= <<= == !=
  - <= >= && || ++ -- , ->* ->
  - () []

Program for illustration of Operator Overloading

```
class FLOAT
{
    float no;
    public:
    FLOAT(){}
    void getdata()
    {
        cout<<"\n ENTER AN FLOATING NUMBER :";
        cin>>no;
     }
     void putdata()
     {
        cout<<"\n\nANSWER IS   :"<<no;
     }
     FLOAT operator+(FLOAT);
     FLOAT operator*(FLOAT);
     FLOAT operator-(FLOAT);
     FLOAT operator/(FLOAT);
};
```

Arithmetic Operators Overloaded

```
FLOAT FLOAT::operator+(FLOAT a)
{
    FLOAT temp;
    temp.no=no+a.no;
    return temp;
}
==========FLOAT
FLOAT::operator * (FLOAT b)
{
    FLOAT temp;
    temp.no=no*b.no;
    return temp;
 }
```

```
FLOAT FLOAT::operator - ( FLOAT b)
{
 FLOAT temp;
temp.no=no-b.no;
return temp;
}
==========
FLOAT FLOAT::operator / (FLOAT b)
{
FLOAT temp;
temp.no=no/b.no;
return temp;
}
```

Operator Overloading (Contd…)

- There are few operator which can not be overloaded.
- **Operator that are not overloaded** are follows
  - scope operator - ::
  - sizeof
  - Period - .
  - ternary operator - ?:
  - typeid() operator

Operator Overloading - Eg

```
class loc
{
   int x, y;
public:
   loc() {}
   loc(int lg, int lt) {
      x = lg; y = lt;
   }
   void show() {
      cout << x << y;
   }
   loc operator+(loc op2);
 loc operator-(loc op2);
 loc operator&&(loc op2);
};
```

```
loc loc::operator+(loc op2){
   loc temp;
   temp.x = op2.x + x;
   temp.y = op2.y + y;
   return temp;
}

loc loc::operator-(loc op2){
   loc temp;
   temp.x = op2.x - x;
   temp.y = op2.y - y;
   return temp;
}

loc loc::operator&&(loc op2){
   loc temp;
   temp.x = op2.x && x;
   temp.y = op2.y && y;
   return temp;
}
```

Operator Overloading - Eg

```
int main()
{
   loc ob1(10, 20), ob2( 5, 30);loc ob3;
   ob1.show();
   ob2.show();
   ob3 = ob1 + ob2;
   ob3.show();
   ob3 = ob1 - ob2;
   ob3.show();
   ob3 = ob1 && ob2;
   ob3.show();

   return 0;
}
```

Operator Overloading (Contd…)

```
loc loc::operator=(loc op2)
{
   x = op2.x;
   y = op2.y;
   return *this;
}
```

Overloading Assignment Operator

```
loc loc::operator++()
{
   x++;
   y++;
   return *this;
}
```

Overloading Unary Operator

Rules of Operator Overloading

1) Only built-in operators can be overloaded. New operators can not be created.

2) Arity of the operators cannot be changed.

3) Precedence and associativity of the operators cannot be changed.

4) Overloaded operators cannot have default arguments except the function call operator () which can have default arguments.

5) Operators cannot be overloaded for built in types only. At least one operand must be used defined type.

Are you ready to solve…

1. The associativity of the operators can be changed if operator is overloaded.
   a. True
   b. False

   Ans: **b. False**

2. Which operator can be overloaded?

   a. . b. :: c. ?: d. None of them

   Ans: **d. None of them**

End of Module 7

Disclaimer

- Some examples and concepts have been sourced from the below links and are open source material
  - **http://cppreference.com**
  - **www.cplusplus.com**

- References:
  - *C++: The Complete Reference* - 4th Edition by Herbert Schildt, Tata McGraw-Hill publications.
  - *The C++ Programming Language*- by Bjarne Stroustrup.
  - *Practical C++ Programming*- by Steve Oualline, O'Reilly publications.

Learning & Development Team

ITPB Road Whitefield
Bangalore 560 048 India
Tel +91 80 2297 9123
Fax +91 80 2841 1474
e-mail info@tataelxsi.com

www.tataelxsi.com