Question No:1

Operator overloading is similar to function overloading, where you have many versions of  
the same function differentiated by their parameter lists. Operator overloading can be done  
using three ways i.e. as a Member function, as a Non-Member function and as a Friend  
function. Answer the following questions regarding operator overloading in C++.  
• What are the rules of Operator Overloading? Please provide a list.  
• Which operators cannot be overloaded ?  
• Which operators can not be overloaded as member function?  
• Why we can not overload << and >> operators as a member function (Remember  
Error during class demo)

• What are the rules of Operator Overloading? Please provide a list.

Answer:

1. Only C++ operators can be we use as an overloaded.
2. Precedence and associativity of the operators cannot be changed.
3. We cannot send the by Default Argument to the Operator Overloaded
4. Operators cannot be overloaded for built in types only. At least one operand must be used defined type.
5. Assignment (=), subscript ([]), function call (“()”), and member selection (->) operators must be defined as member functions
6. All other operators can be either member functions or a nonmember functions.

• Which operators cannot be overloaded?

1. :: Scope Resolution Operator
2. .\* Pointer to Member Function
3. . Dot operator
4. ? Conditional Operator
5. Sizeof operator connot be used
6. Tyeiped Function

Which operators cannot be overloaded as member function?

Insertion operator (<<) and Extraction operator (>>) cannot be overloaded in the member function.

Why we cannot overload << and >> operators as a member function?

As they are defined as friend function in the standard library of C++ “using namespace std;” that’s why they can’t be overloaded in member function.