Operator Overloading

Q What is OO?

A. OO means to create multiple use for operators like +, -, =, pre-inc, post-inc, <<, >>

- There operators can be used to operate with objects. For eg

```
obj1+ obj2;
obj1 = obj2;
Cout << obj1; etc.
```

Q Why is OO done?

A. So that these operators can be used with objects.

For eg

```
obj1+ obj2;
obj1 = obj2;
```

Q How is it done?

A OO is done by writing operator functions for each operator.

• For eg :- To overload + operator :-

Student Stu1t :: operator + (Stu1t Po1)

Q In what ways oo can be done?

A oo can be done thru 1) member fun 2) Friend fun.

Q what are the rules for OO?

A 1) Atleast, one operand must be a class obj.

eg obj1 + obj2;
obj1 + 10;

2) Only existing operators can be overloaded

3) These opr can't be overloaded

?::
sizeof
::
.
.*
(cont)

4) You can't rename existing opr.

5) You can't change No. of operands.

6) You can't change precedence & associativity.

Q Wh. opr can't be overloaded thru MF?

A >> and <<

Q Wh. opr can't be overloaded thru FF?

A = operator.

Q WAP to overload +, =, pre-inc, post-inc thru Member fun for student class

A see codes

Q WAP to overload +, pre-inc, post-inc, << and >> thru FF for Stud. class?

A see codes.