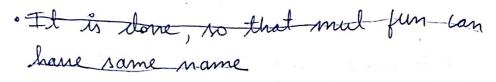
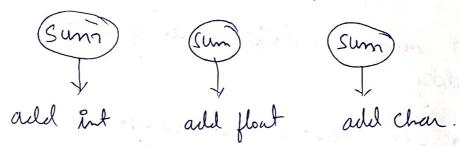
Function Overloading in C++ 0/L = overloading // Short terms Polymorphism Operator 0/23 fun 0/Lg · Polymorphism is a general term Poly means multiple marshim " shape. · PM means mutliple une aj same entity . In C++, there & two ways to implement PM - fun 0/Lg - Operator 0/L9. a what is fun o/L3? Why is it done? It means, there are several fun with same name. FOL is only allowed in C++ but not in C

1



· It is done to allow having same name for different fun. So, that user doen't need to have different name for similar fun.

Suppose, there & It digerent fun for adding int, char, floats



· User can use same name for 3 different fun.

Duhat is the rule for FOL?

A: Different fun can have same name

\*\*But barameters should be different

- No. of barameter should be different

- type of " can ";

eg void sum (int 4, int y); // 2 param of type int Void Sum (it x, it y, int 3); 1/3 param Veid sum (char x, mt y): 1/2 param of type int of types Chan and ent. These v 3 overloaded fun

with different param.

EXAMPLE of FOL-1 Q do not use class. 0/L a sum fun for - 2 int - 1 char, 1 int

Answer #include (iosteam) ving namespare std; Your sure Swar ( in the state // Define 3 sum fun w/o any class. void sum (int x, int y) 1/sum fun int add = x+y; cont << add;

```
// Define sum with 3 int
Void sum (it x, int y, int 3)
    int add = x + y + 3;
     Cout Kadd;
1/Define sum with 1 char, 1 int
 void sum (chan x, int y)
      char add = x +y;
      Loute add;
 int main ()
    int a=10, b=20, c=30')
     Chan ch = (a);
     Sum (a, b) -, // 0/p = 30
     sum (a, b, c); 110/p=60.
  Sum (ch, 1), //0/P = 6
    // Ch is 'a'. I will be added
    to assir value of 'a' which
    will become arein value of 'b'
    So O/P will be b.
```

```
Example of FOL-2
a overload a class member fun.
 . overload the set fun in student class
  Create ? set fun
         1) with 3 parameter
        2) with 1 paramter. It will
          only assign value to rn.
    # include (instrem >
     ching homespace std;
      Class student
         private:
         int m, m1, m2;
        public:

// set fun w/ 3 Param.

veid set (it *, it y, it z);
         Manother set fun w/ 1 param
         veil set (int x);
         void get ();
    void student: set (it 1, it y, it 3)
          Yn=x'
          m1=4;
          m 2 = 3 .
```

```
// Define the second set fun
void student:: set (it n)
      11 It will arrigh values
        + only rn;
void studt: get ()
int main ()
       Studt obj1, 05,2%
       obj 1. set (5, 50,50); //rn=5.
      obj ] · get () ;
      obj 2. set (2, 30,30),
       0 bj 2 - get ()
       obj 1. set (1): // change rn of obj 1.
       Marsign rn=1 to 0671
      obj1. get (),
       0/P: -
       5,50,50.
       2,30,30
       1,50,50.
```

## Constructor Overloading

Quhat is constructor overlanding? Why is it done?

A · Constructor is a type of fun.

· COL means to create multiple

constructors. in a class.

. It is done to have flexibility while creating objects of a class

Eg If there is only I comt. in class then obj can be created only by parsing values. We can't create obj w/o parsing values.

1) student 06; 1; // There will be an enor, 6 coz there is a parameterized court. in dan.

Destudent obj 1 (1,10,20), 11 This is correct.

Values shud be parsed while creating obj.

In this ego, there is only 1 comf.

So obj can only be created by parsing values.

Values.

- We have no option of creating obj w/o Parsing values.

7

· Now with the help of COL, We can have flexibility to create 05) both ways - with passing value and withollt paring values.

( How is to COL derie? · WAP with 2 court. Creste a

student class

· Create 2 Court 1) with parameter

2) W/o Param. i.e a blank Comt.

· Now, create 05° in two ways 1) By passing values Student obj! (1,10,20); // Parameterzed court Called.

2) by W/o Parsing Values student obj 2; // Blank Count.

# inclu - - -. Using - - -.

clas student

int rn, m1, m2; public:

Void set (it 4, it 7, it 3); void get (); Student ( int in, it y, it 3); [[ comst. student (); ]] No param. Param

```
// In the previous class, two const
are declared:
         1) Parameterized Court
         2) Non -Parameterized const.
       studt: set (it u, it y, it 3)
        m1=4
void student : : get ()
        Contecneemizemz;
 tou 11 Define Parameterized count.
 spend studi Studt (it 7, it 7, I 3)
         coutéé " const Called";
  // Define non-parameterized const.
student:; Student ()
          Cout << " Cont Called";
```

int main()

Student obj! (1,10,20), obj2;

// obj 1 will call parameterized cont

1/obj2 11 1. non parameterized

const.

Obj 1. get ();

Obj 2. set (2,22,22);

Obj2. get ();

Oly 2. set (2,22,22);

2,22,22.

oj

Polymorphism

fun 0/L Operation 0/L.

Eg 1:-0/L sum fun three times

Eg 2:-0/L set fun in student class

2 times.

Eg 3: Court. Overloading-Overload the Court 2 times.