Object Passing

Q What does OP mean?

A OP means to pass obj as parameter / argument to fun.

For eg `obj1.set(10,30);`

integers passed.

`obj1.add_obj(obj2, obj3)`

obj passed to the fun.

Q Why do we need OP?

A To perform inter-object operations.

For eg adding var of two obj
Printing var of two obj
etc.

Q How is obj passing done?

A The fun call is done this way

`obj3.add_obj(obj2, obj1);`

↓

Calling obj Passing obj.

Q Which obj shud be calling obj?

A The calling obj shud always be the one in which you need to store or modify values.

Q Inside a fun, if we modify the passing obj, will the changes in the passing " , be permanent?

A No, changes made to passing obj r not permanent. So, changes shud always be made to the calling obj.

Steps to create a fun with OP

```
Class student
{
    int rn, m1, m2;
    Public:
     void set(---);
     void get();
     void add();
     void add_obj(student p01;
                  student p02);
};
```

You need to add this fun in class.

```
void student :: add_obj (student obj1,
                         student obj2)
{
   ---
   ---
   ---
   ---
}
```

You need to write this fun definition.

Logic to write the fun Def

1) Start with fun call.
   Write the fun call first.

```
obj3.add(obj1, obj2);
//We want to add obj1 & obj2 & copy
  to obj3.
// So obj3 will be the calling obj.
```

```
obj3.add_obj(obj1, obj2);
```

obj3 → Calling obj

obj1, obj2 → Pass obj

2) Change names of obj1 & obj2

obj3.add_obj(obj1, obj2)

↓ ↓

P01 P02

3) Write variables of P01 & P02.

obj3.add_obj(obj1, obj2)

↓ ↓

P01 P02

↓ ↓

rn	P01.rn	P02.rn
m1	P01.m1	P02.m1
m2	P01.m2	P02.m2

4) add var of P01 & P02 & copy to obj3 var

m1 = P01.m1 + P02.m1;

m2 = P01.m2 + P02.m2;

5) How to write fun declaration

Take eg. of set fun:

void student::set(int x, int y, int z);

↓ ↓ ↓ ↓ ↓

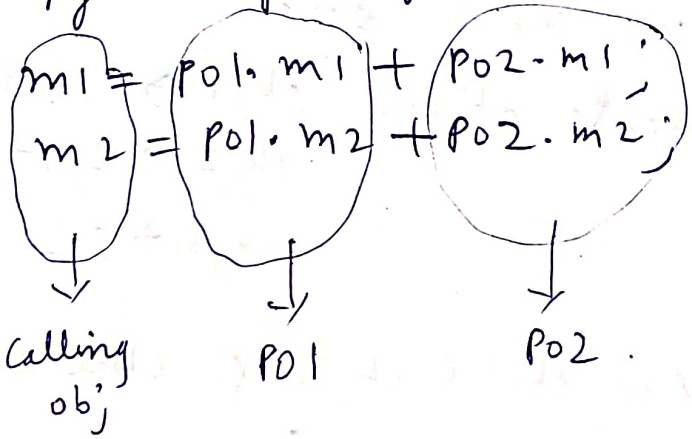
void student::add_obj(student P01, student P02);

↑ Change fun name

↑ • Put two parameters
• Instead of int x, y pass objects P01, P02.

6) How to write fun definition

```
void student::add_obj (student P01,
                       student P02)
{
    //Copy the part from step-4
    m1 = P01.m1 + P02.m1;
    m2 = P01.m2 + P02.m2;
}
```



Q The complete code for obj pass?

- Create a class student.
- Write a fun add_obj( ) to add two obj of this class & add to the third obj.
- Create objects obj1 → 1, 3, 4.
  obj2 → 2, 97, 96
  obj3 → 3, -1, -1.
- Add obj1 & obj2 & store in obj3.

A

```
#include <iostream>
Using namespace std;
Class student
{
        private:
        int m, m1, m2;
        Public:
        void set (int x, int y, int z);
        void get ();
        void add ();
        void add_obj (student po1,
                      student po2);
    //This fun shud take two obj
      as parameters
};
```

```
void student :: set (int x, int y, int z)
{   rn = x;
    m1 = y;
    m2 = z;
}
void stud :: get ()
{     cout << rn << m1 << m2;
}
void stud :: add ()
{     cout << m1 + m2;
}
void student :: add_obj ( student P01,
                          student P02 )
{     m1 = P01.m1 + P02.m1;
      m2 = P01.m2 + P02.m2;
} // Steps to write this fun explained
     earlier
int main ()
{   student obj1, obj2, obj3;
    obj1.set (1, 3, 4); // assign values
    obj2.set (2, 97, 96);
    obj3.set (3, -1, -1);
    obj3.add (obj1, obj2); // add obj1 & obj2 to obj3.
    obj3.get (); // Print obj3
}
```

O/P

3, 100, 100