

4. Abstraction: Abstraction is another very important principle of OOP and it says that we must hide the implementation details from the user and should only expose the necessary components.

For ex: To use a Television we are only required to know how to use its remote and not the internal working of that television i.e how the circuitry is working, how the picture is getting delivered and how the channels are communicating with their respective servers.

The main benefit of abstraction is that the end user of our product can easily interact and use it without getting involved in its complexities.

Creating Parametrized Member Fn

```

#include <iostream.h>
#include <string.h>
#include <conio.h>
class Emp
{
    int age;
    char name[20];
    float sal;
public:
    void set (int, char *, float);
    void show ();
};

void Emp::set (int a, char *p, float s)
{
    age = a;
    strcpy (name, p);
    sal = s;
}

void Emp::show ()
{
    cout << age << " " << name << " " << sal << endl;
}
    
```

*format of parametrized member function*

E (2613)	
age	25
name	"Rahul"
sal	30000.0

F (2613)	
age	30
name	"Amit"
sal	28000.0

```

int main()
{
    clrscr();
    Emp E, F;
    E.set(25, "Rahul", 30000.0);
    F.set(30, "Amit", 28000.0);

    E.show();
    F.show();
    getch();
    return 0;
}

```

*actual arg*