

# OBJECT-ORIENTED PROGRAMMING

## Operator overloading

**Le Van Vinh, PhD**

Faculty of Information Technology

University of Technology and Education

Email: [vinhlv@fit.hcmute.edu.vn](mailto:vinhlv@fit.hcmute.edu.vn)

## An example

- ❖ `PhanSo ps1=new PhanSo();`
- ❖ `PhanSo ps2=new PhanSo();`
- ❖ `PhanSo kq=ps1.Add(ps2);`
  
- ❖ How can we do?
- ❖ `PhanSo kq=ps1 + ps2;`

## An example

- ❖ `PhanSo ps1=new PhanSo();`
- ❖ `PhanSo ps2=new PhanSo();`
- ❖ `if(ps1.SoSanh(ps2) == true)`  
    `Console.WriteLine("ps1 > ps2");`
- ❖ How can we do?
- ❖ `if(ps1 > ps2)`  
    `Console.WriteLine("ps1 > ps2");`

# What is operator overloading?

- ❖ Operator overloading provides a much more natural way of implementing the operations on custom types

# Binary operators

- ❖ Binary operators function will take 2 arguments and return a new object of the Containing type
- ❖  $+, -, *, /, \%, >, >=, <, <=, ==, !=, \dots$
- ❖ <https://docs.microsoft.com/en-us/cpp/cpp/binary-operators>
- ❖ Xem ví dụ operator + trên lớp Phân số

# Unary operator

- ❖ A function taking only one argument of the containing type
- ❖ ++, --, !,
- ❖ <https://docs.microsoft.com/en-us/cpp/cpp/overloading-unary-operators>
- ❖ Xem ví dụ operator ++ trên lớp Phân số.

# Static classes/class members

```
namespace System
{
    ...public static class Math
    {
        ...public const double PI = 3.1415926535897931;
        ...public const double E = 2.7182818284590451;

        ...public static decimal Abs(decimal value);
        ...public static double Abs(double value);
        ...public static float Abs(float value);
        ...public static int Abs(int value);
        ...public static short Abs(short value);
        ...public static sbyte Abs(sbyte value);
        ...public static long Abs(long value);
        ...public static double Acos(double d);
    }
}
```

# Static classes/class members

- ❖ A **static class** cannot be instantiated
- ❖ The **static member** is callable on a class even when no instance of the class has been created. The static member is always accessed by the class name, not the instance name
- ❖ **NhanVien.TinhLuong();**