

OOP or FP?

To be an OOP Language

- You need to have **classes**
- Encapsulation, Inheritance, Polymorphism

To be a FP Language

- You need to have **function** pointers
- Function types

Why choose?

- Most languages support **both**

FP in Java

Pointing to a method

```
person.java

public class Person {
    private String name;

    public Person(String name) {
        this.name = name;
    }

    public void greet(String timeOfDay) {
        System.out.println("Good " + timeOfDay + ", " + this.name);
    }
}
```

```
program.java

public class Program {
    public static void main(String[] args) {
        Person person1 = new Person("Alice");
        Person person2 = new Person("Bob");

        Consumer<String> greeter1 = person1::greet;
        Consumer<String> greeter2 = person2::greet;

        greeter1.accept("morning"); // Output: Good morning, Alice
        greeter2.accept("evening"); // Output: Good evening, Bob
    }
}
```

FP in C#

Pointing to a method

C# person.cs

```
public class Person
{
    private string name;

    public Person(string name)
    {
        this.name = name;
    }

    public void Greet(string timeOfDay)
    {
        Console.WriteLine($"Good {timeOfDay}, {this.name}");
    }
}
```

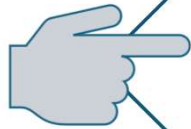
C# program.cs

```
public class Program
{
    public static void Main(string[] args)
    {
        Person person1 = new Person("Alice");
        Person person2 = new Person("Bob");

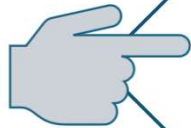
        Action<string> greeter1 = person1.Greet;
        Action<string> greeter2 = person2.Greet;

        greeter1("morning"); // Output: Good morning, Alice
        greeter2("evening"); // Output: Good evening, Bob
    }
}
```

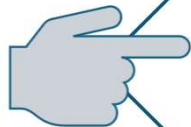
Function Pointers in OOP Languages



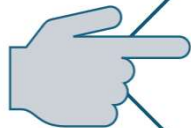
Point to **function** and
object



Are called **delegates**



Are executed in the **context**
of a specific **object**



Are **objects** themselves...