

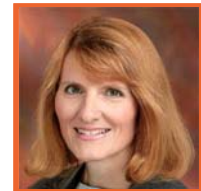
# Final Words and Next Steps

Deborah Kurata

<http://msmvps.com/blogs/deborahk/>

@DeborahKurata

deborahk@insteptech.com

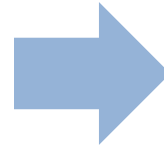


**pluralsight**  
hardcore dev and IT training



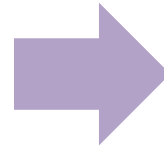
# Object-Oriented Programming (OOP)

## Identifying Classes



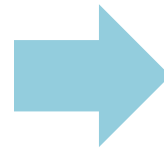
- Represents business entities
- Defines properties (data)
- Defines methods (actions/behavior)

## Separating Responsibilities



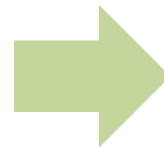
- Minimizes coupling
- Maximizes cohesion
- Simplifies Maintenance
- Improves Testability

## Establishing Relationships



- Defines how objects work together to perform the operations of the application

## Leveraging Reuse



- Involves extracting commonality
- And building reusable classes/components

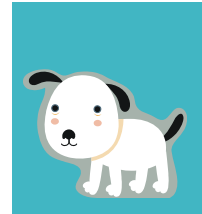
# Four Pillars of OOP



# Abstraction

## Abstraction

- Simplifying reality
- Ignoring extraneous details
- Focusing on what is important for a purpose



Joe Smith  
Joe@aol.com  
123 Main St.

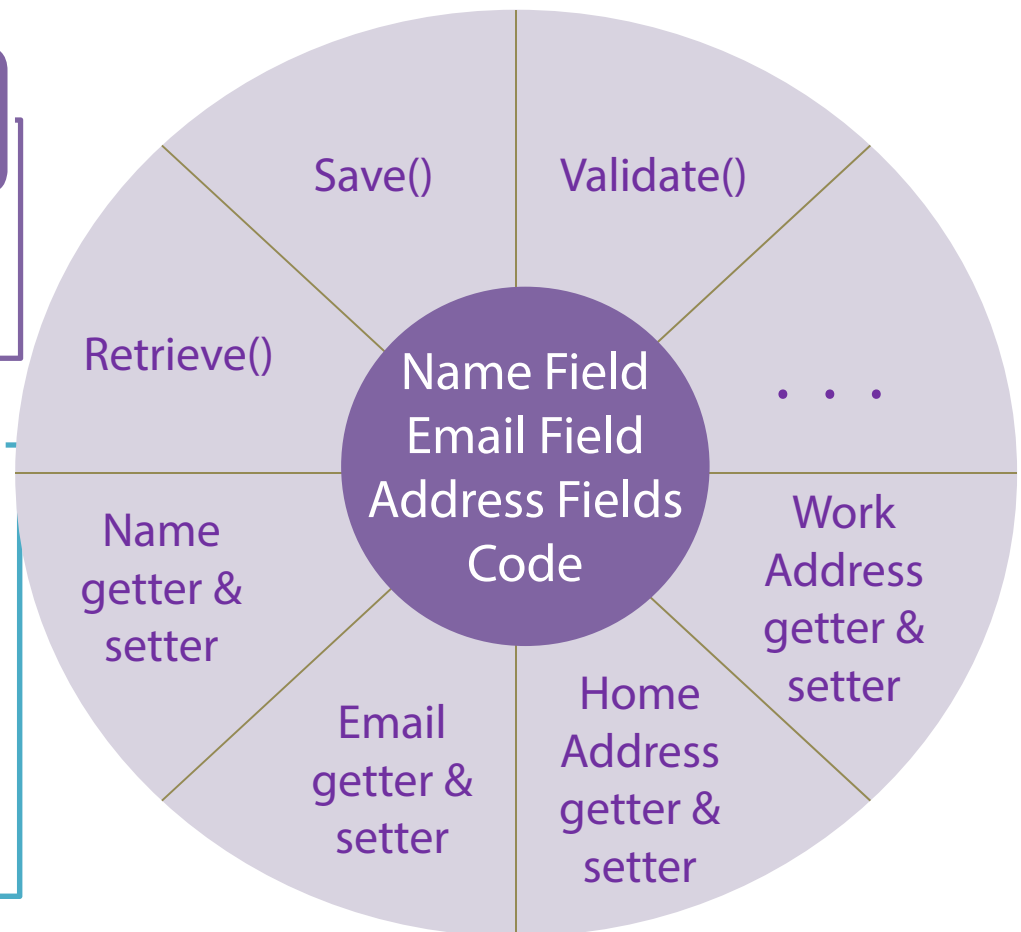
# Encapsulation

## Data hiding

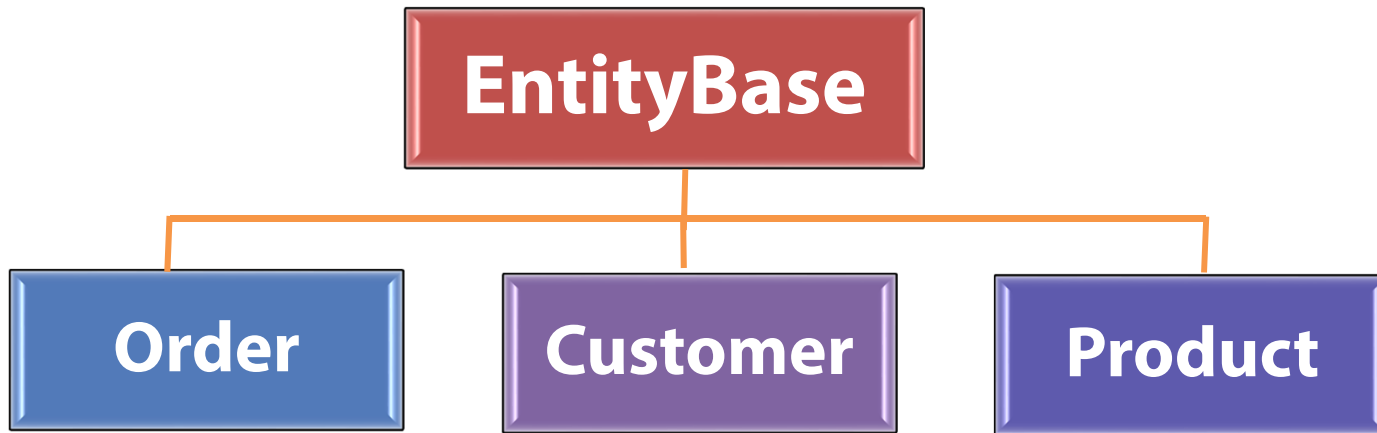
- Protects the data

## Implementation hiding

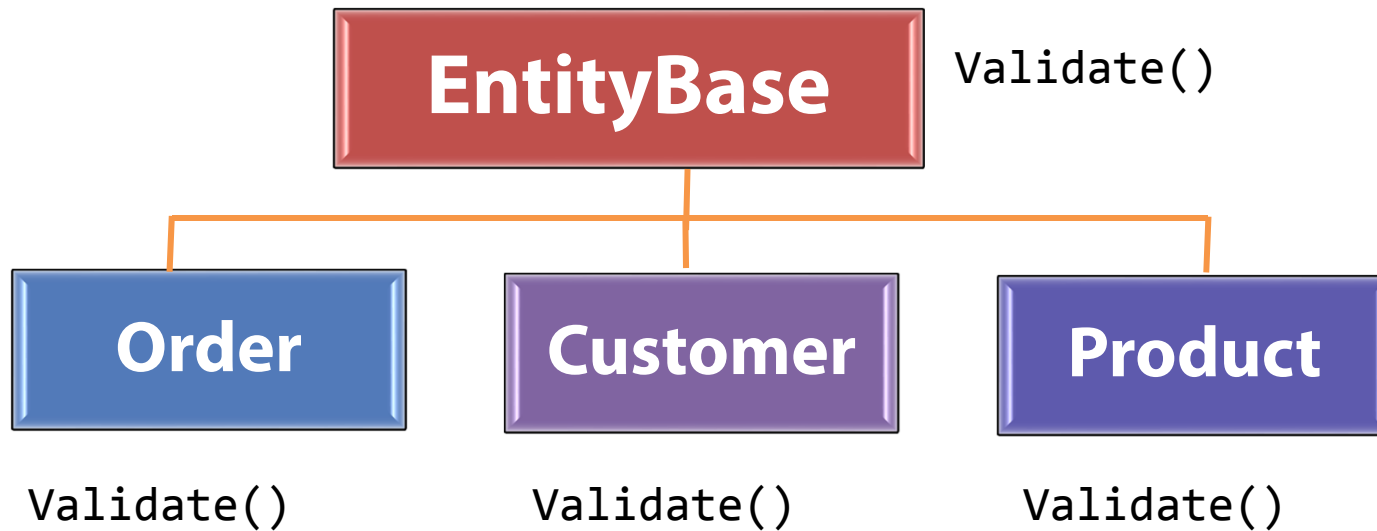
- Helps manage complexity
- Implementation can be changed without impacting the application



# Inheritance

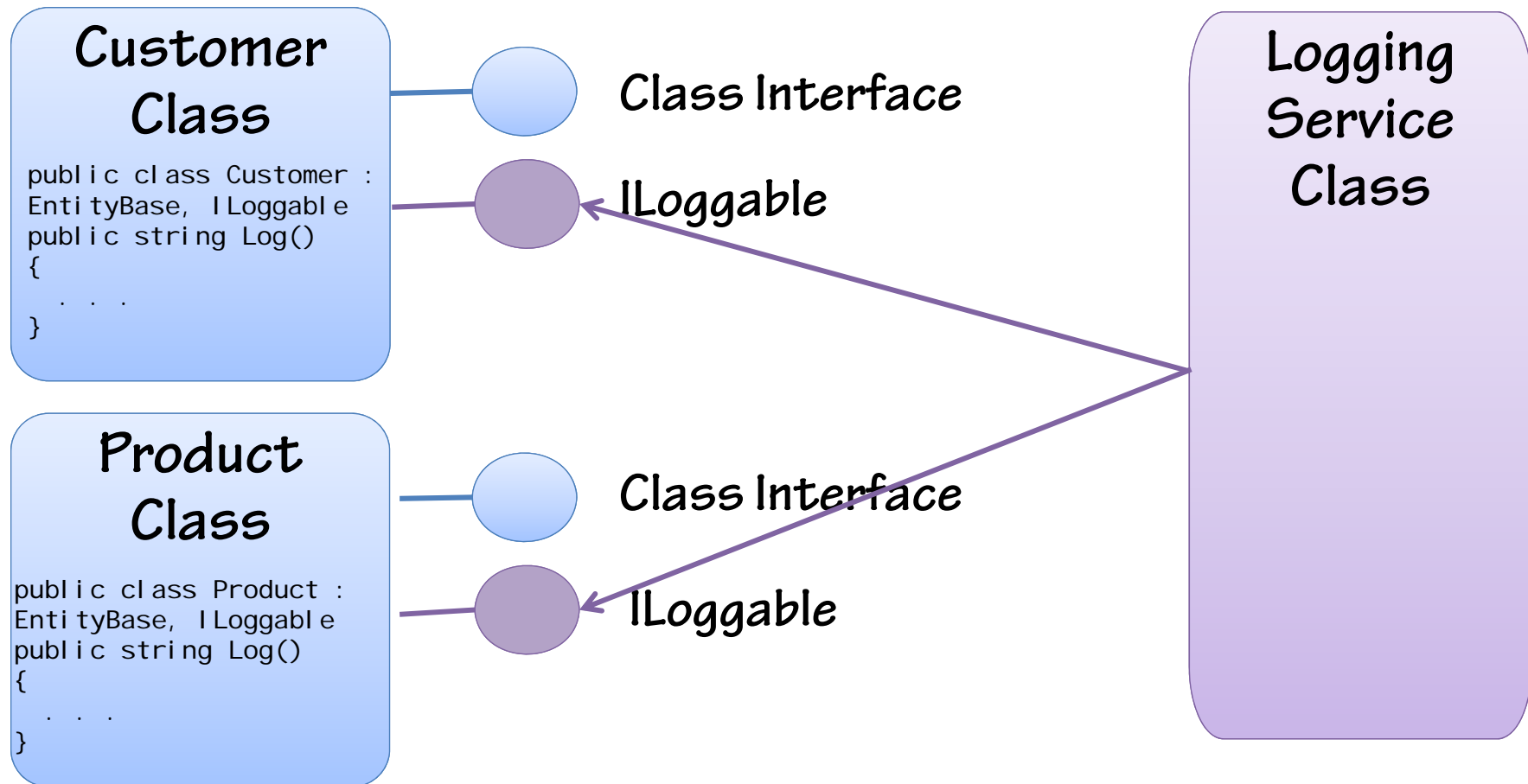


# Inheritance-Based Polymorphism



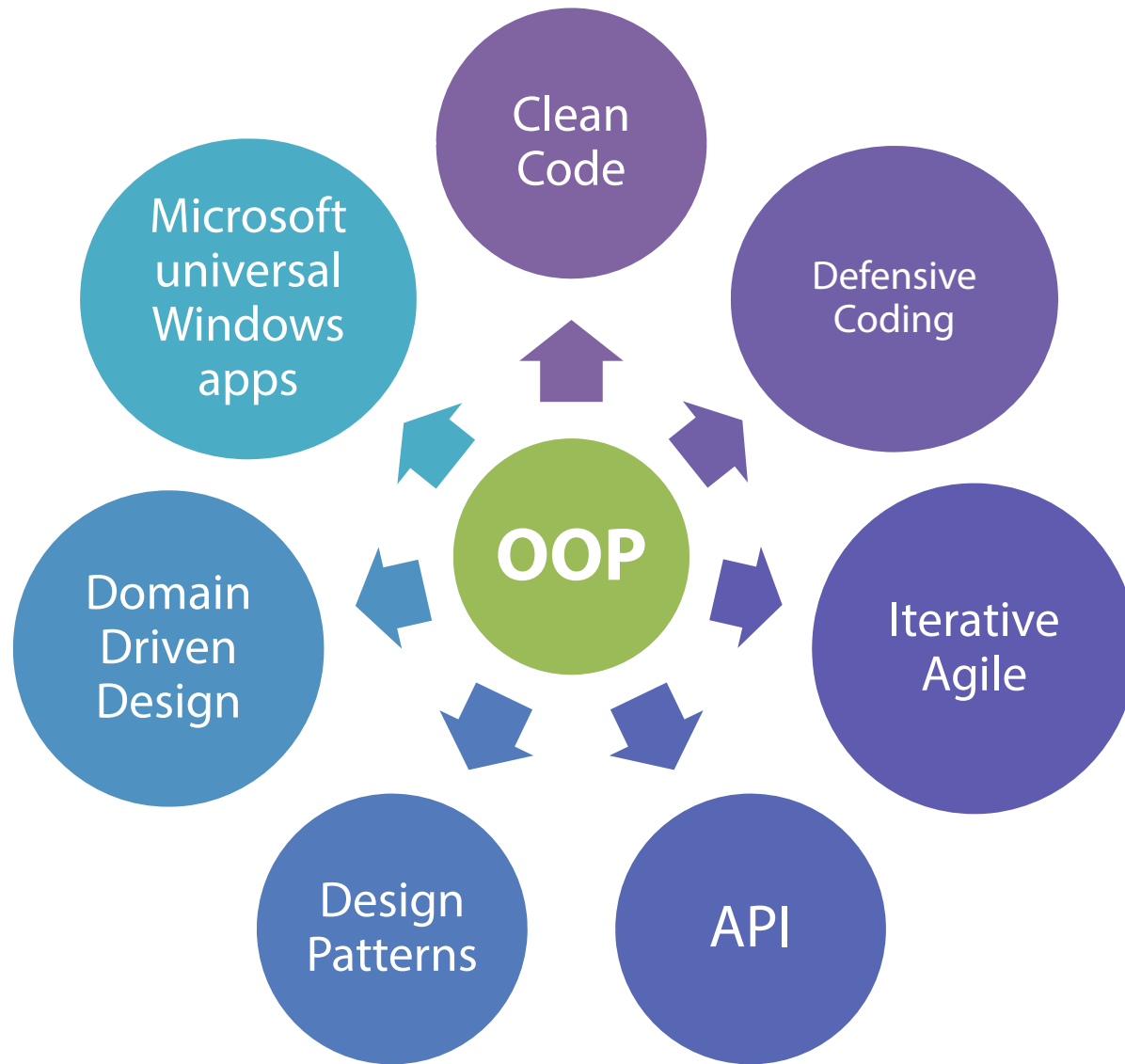
# Interface-Based Polymorphism

```
public interface ILoggable  
string Log()
```

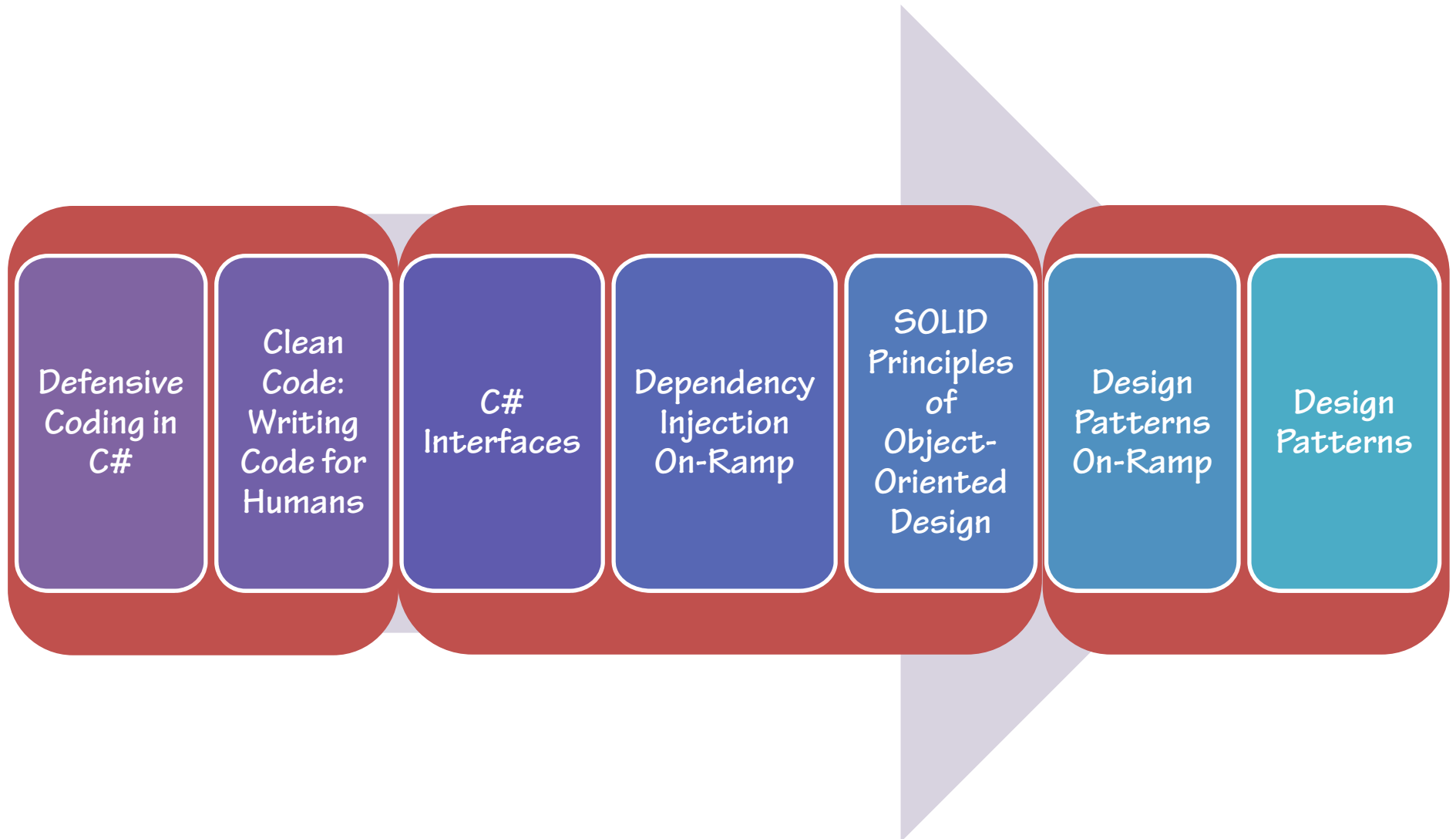




# OOP is the Foundation



# Learning Path



# Four Pillars of OOP

