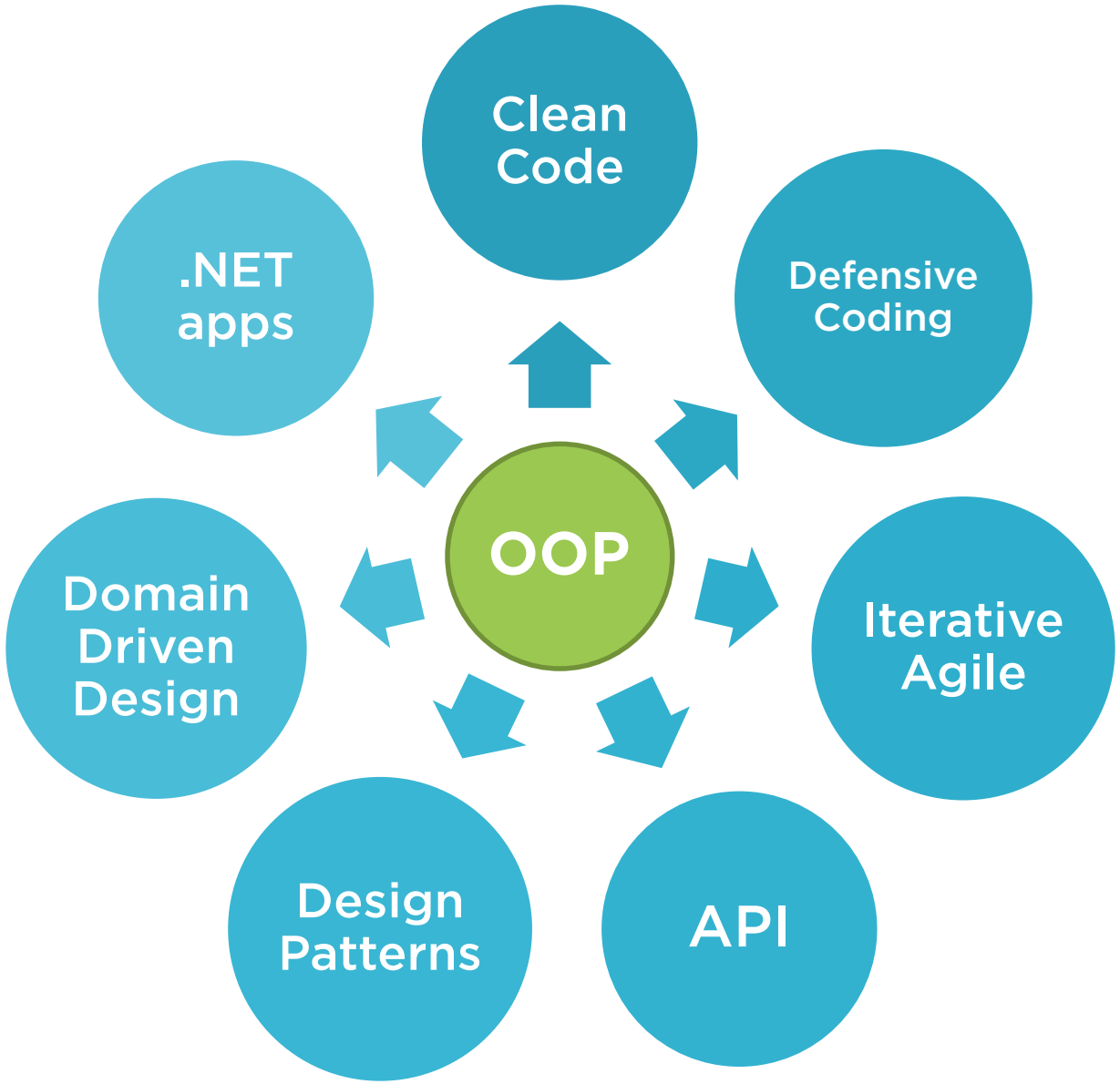
Object-Oriented Programming Fundamentals in C#

INTRODUCTION

Deborah Kurata
CONSULTANT | SPEAKER | AUTHOR | MVP | GDE

@deborahkurata | blogs.msmvps.com/deborahk/

OOP Is the Foundation



Object != Class

Properties

Methods

```
public class Customer
{
    public int CustomerId { get; set; }

    public string EmailAddress { get; set; }

    public string FirstName { get; set; }

    public string LastName { get; set; }

    public bool Validate(){ ... }
}
```

Object != Class

Members

```
public class Customer
{
    public int CustomerId { get; set; }

    public string EmailAddress { get; set; }

    public string FirstName { get; set; }

    public string LastName { get; set; }

    public bool Validate(){ ... }
}
```

Object != Class

```
var customer = new Customer();
```

Object variable

```
public class Customer
{
    public int CustomerId { get; set; }

    public string EmailAddress { get; set; }

    public string FirstName { get; set; }

    public string LastName { get; set; }

    public bool Validate(){ ... }
}
```

Objects

Class

We need to define the business objects

```
public class Customer
{
    public int CustomerId { get; set; }

    public string EmailAddress { get; set; }

    public string FirstName { get; set; }

    public string LastName { get; set; }

    public bool Validate(){ ... }
}
```

Business Object == Class


Entity

Customer Management System

Entity — customer

Class — Customer
- Last Name
- First Name
- Go On An Adventure

Objects — Bilbo Baggins

Frodo Baggins

Object-Oriented Programming (OOP)

An approach to designing and building applications that are:

- Flexible
- Natural
- Well-crafted
- Testable

by focusing on objects that interact cleanly with one another

Identifying classes

Separating responsibilities

Establishing relationships

Leveraging reuse

Prerequisites

Required

- Basic knowledge of C#
  - Syntax
  - Variables
  - Conditionals
  - Control flow

Suggested

- Visual Studio

Not Required

- OOP experience

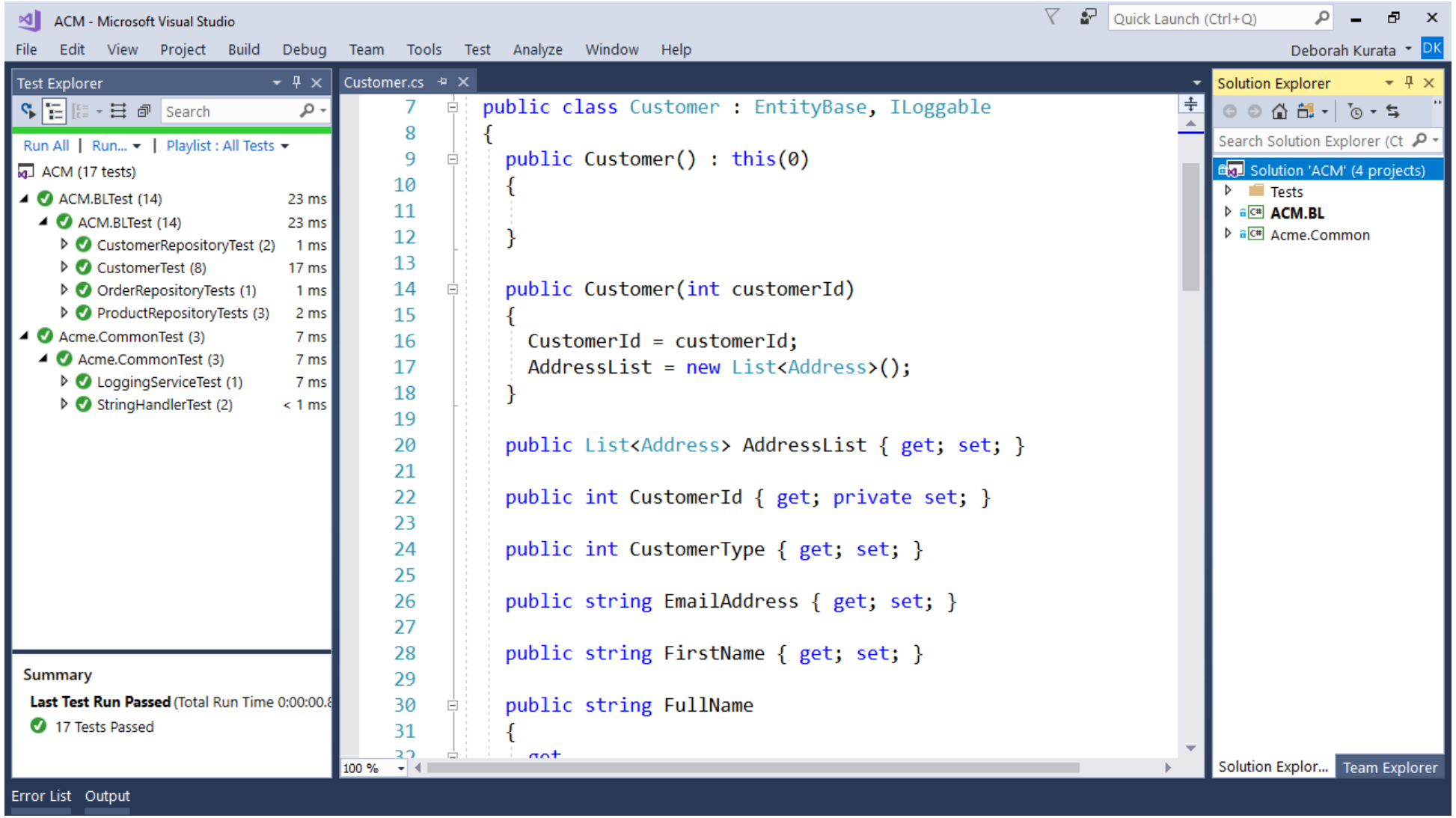
Checklist

Review module concepts

Code along assistance

Revisit as you implement your applications

Coding Along



Course Outline

Identifying classes

- Identifying classes from requirements
- Building entity classes

Separating responsibilities

- Separating responsibilities

Establishing relationships

- Establishing relationships

Leveraging reuse

- Leveraging reuse
- Building reusable components
- Understanding interfaces