MUST

Wisdom & Virtue

MIRPUR UNIVERSITY OF SCIENCE AND TECHNOLOGY
DEPARTMENT OF SOFTWARE ENGINEERING

Object Oriented Programming

Lecture 6: Encapsulation in OOP

Engr.Saman Fatima
Lecturer

Last Lecture

- Abstraction in OOP
- Abstraction at Design Level
- Abstraction at Programming Level (Information Hiding)

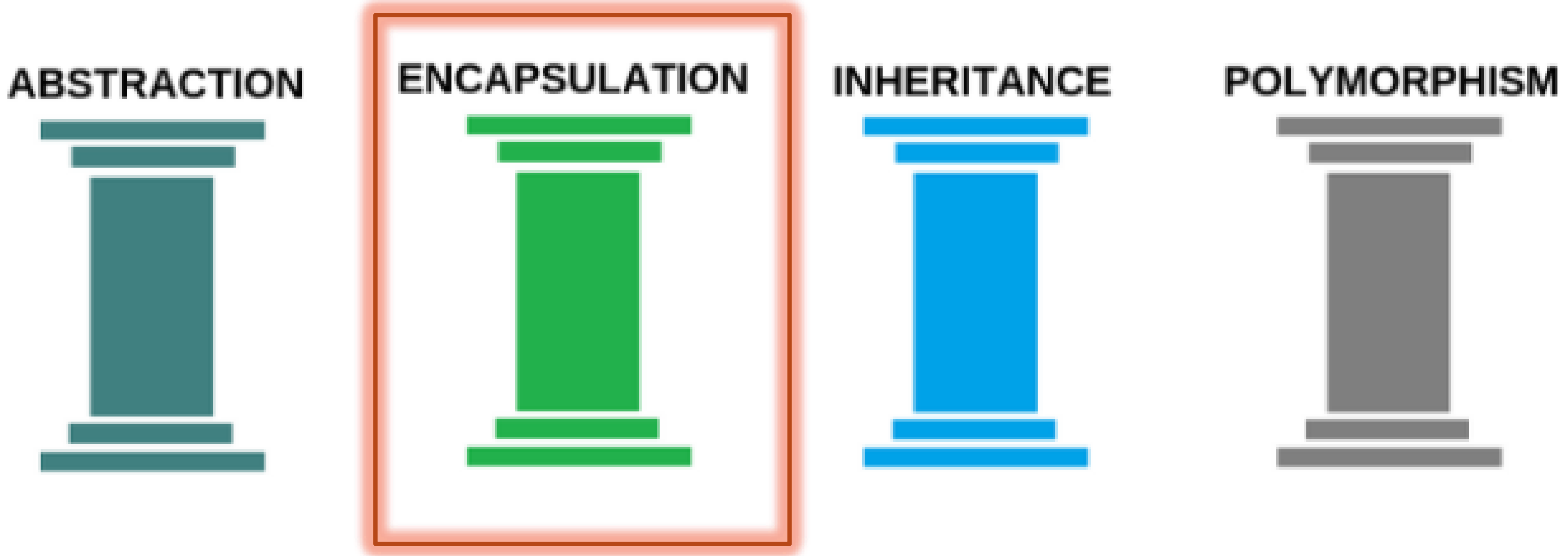
This Lecture

- Encapsulation in OOP
- Accessor functions
- Avoiding Error using Accessor functions
- Read-only Property

Constructor Overloading Graded Task

Create a class named Student with two fields: name and age.
Write three constructors: a default one, one with only name, and one with name and age.
Add a method DisplayInfo() to show student details.
Create three different objects using all three constructors.
Display the output using Console.WriteLine() and keep the console open with Console.ReadLine().

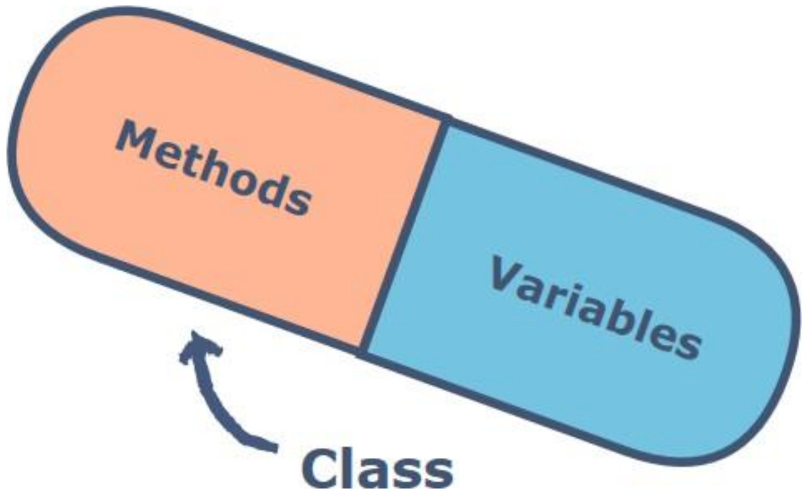
Four Pillars of OOP



Encapsulation in OOP

Encapsulation

- Wrapping up data members and methods together into a single unit (in other words class) is called **Encapsulation**.
- Encapsulation is like enclosing in a capsule.
- That is enclosing the related operations and data related to an object into that object.



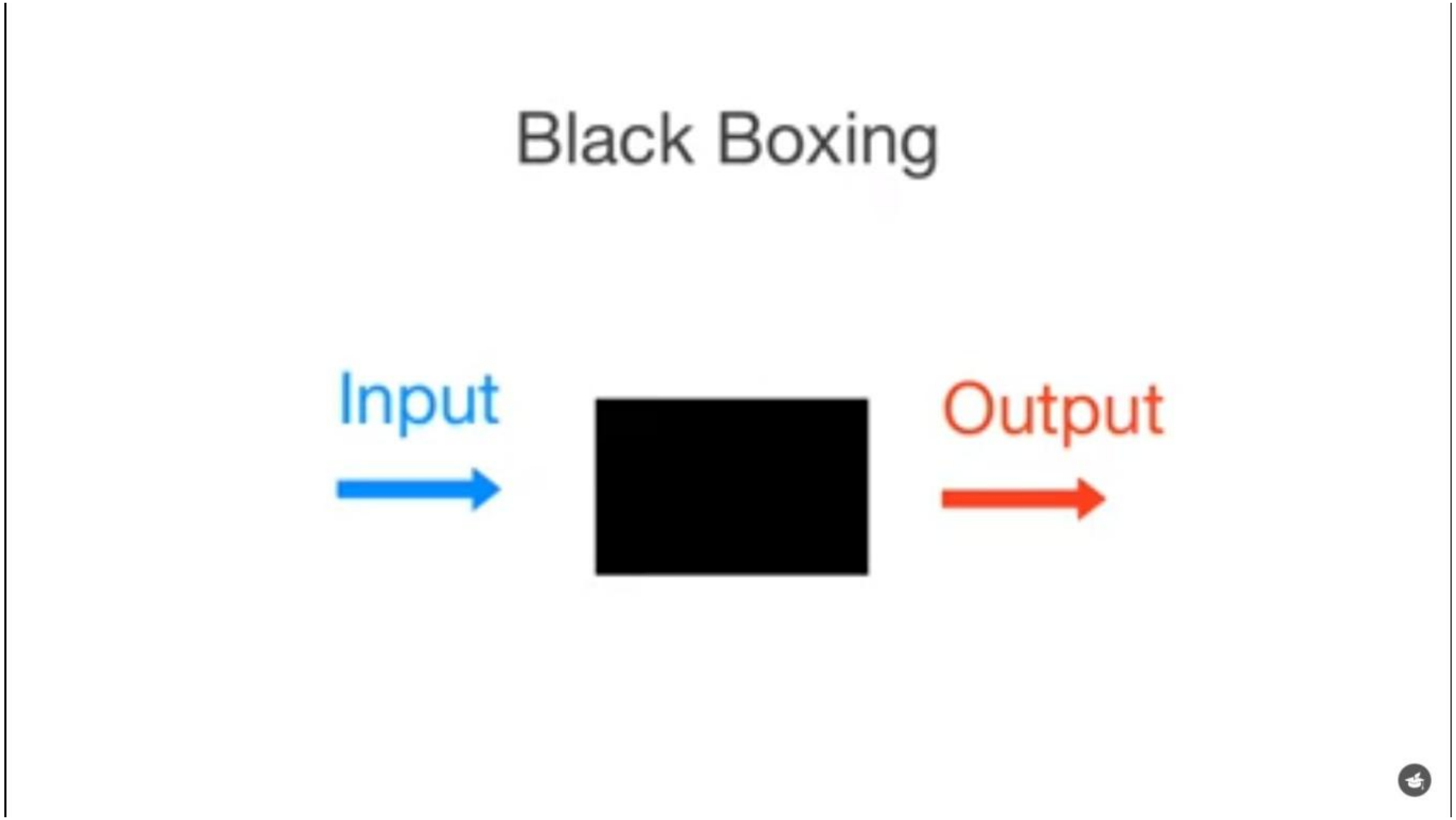
Principle of Encapsulation

Hide all details of the class and let object interact with class **interface** only

Principle of Encapsulation

- **Hide** all details of the class
  - *Make all the fields private*
- Let object interact with class **interface** only
  - *Expose only methods (Interact using public methods)*

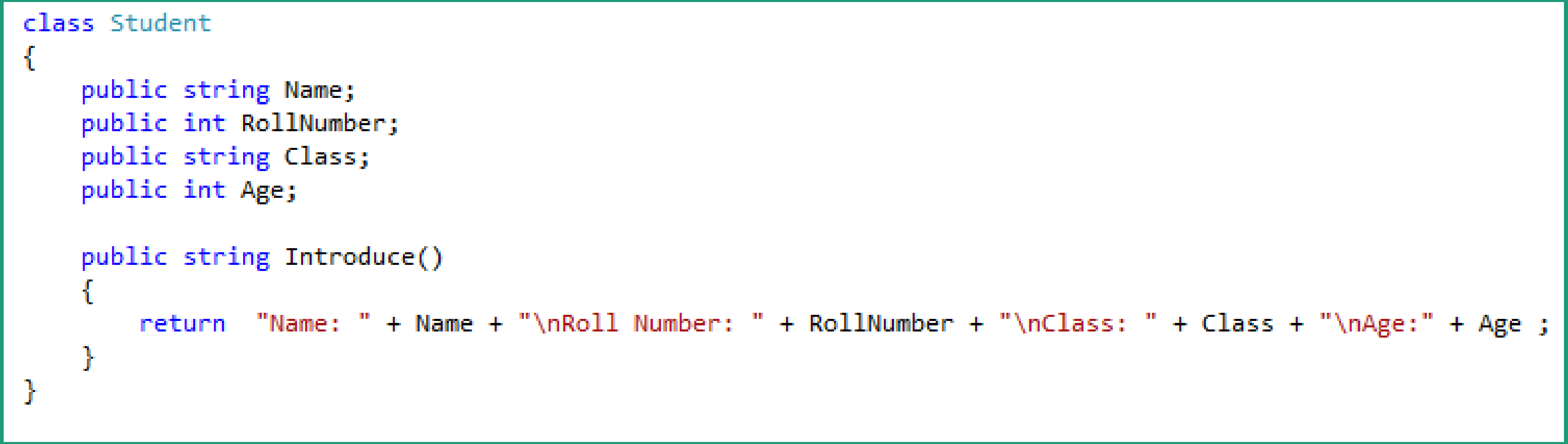
Principle of Encapsulation



Encapsulation

- Encapsulation means binding data and functions into a single unit (class).
- It hides the internal details and only exposes necessary parts through methods.
- Think of a bank account — you don’t directly access balance, you use deposit/withdraw methods.

Student Class



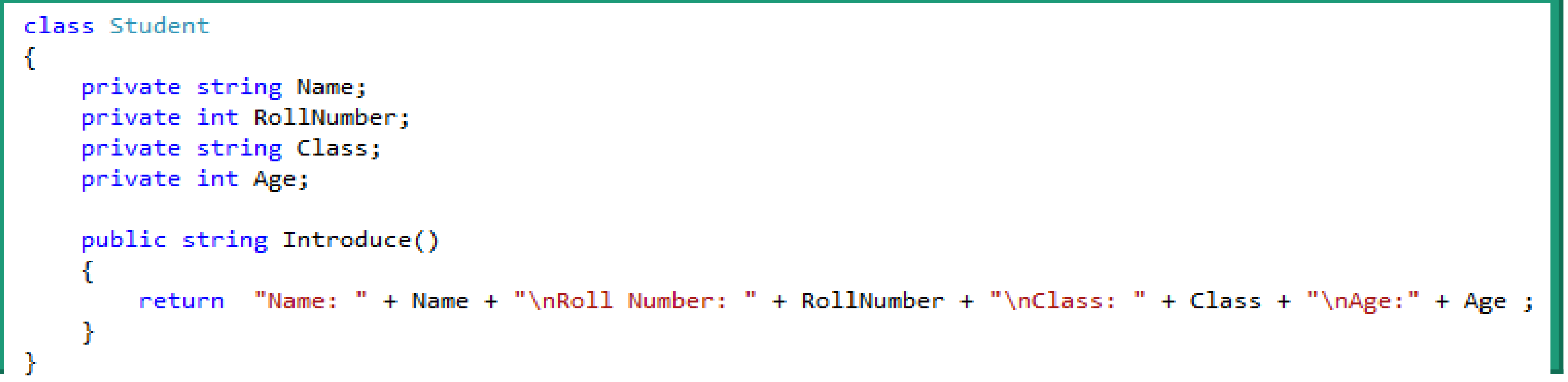
```
class Student
{
    public string Name;
    public int RollNumber;
    public string Class;
    public int Age;

    public string Introduce()
    {
        return  "Name: " + Name + "\nRoll Number: " + RollNumber + "\nClass: " + Class + "\nAge:" + Age ;
    }
}
```

Principle of Encapsulation

Make all fields Private and interact using public function

Encapsulated Student Class



```
class Student
{
    private string Name;
    private int RollNumber;
    private string Class;
    private int Age;

    public string Introduce()
    {
        return  "Name: " + Name + "\nRoll Number: " + RollNumber + "\nClass: " + Class + "\nAge:" + Age ;
    }
}
```

Accessor functions are used to access private data of the object

Accessor Functions(Properties in C#)

- In accordance with principle of **encapsulation** data members of a class are declared as **private** so that outside world can not access the **private data** of the object only **an interface** is provided to outside world in the form of functions.
- Accessor functions are used to access private data of the object, we provide accessor functions to get and set private data members of the class.
- We also add error checking code in accessor functions to reduce errors so that object doesn’t move in illegal state.

Accessor Functions(Properties in C#)

```
private int RollNumber;

public int ROLLNUMBER
{
    get
    {
        return RollNumber;
    }

    set
    {
        RollNumber = value;
    }
}
```

Set Value

```
Student student1 = new Student();
student1.ROLLNUMBER = 10;
```

Get Value

```
Student student1 = new Student();

int RollNumber = student1.ROLLNUMBER;
```

C# Encapsulation Syntax

```
class Student
{
    private string name;  // private field

    public void SetName(string newName) // setter method
    {
        name = newName;
    }

    public string GetName() // getter method
    {
        return name;
    }
}
```

Example in C#

```
using System;

class Program
{
    static void Main()
    {
        Student s1 = new Student();
        s1.SetName("Samna");
        Console.WriteLine("Student Name: " + s1.GetName());
    }
}
```

How to add Properties in a class using class diagram

Avoiding Error using Accessor functions

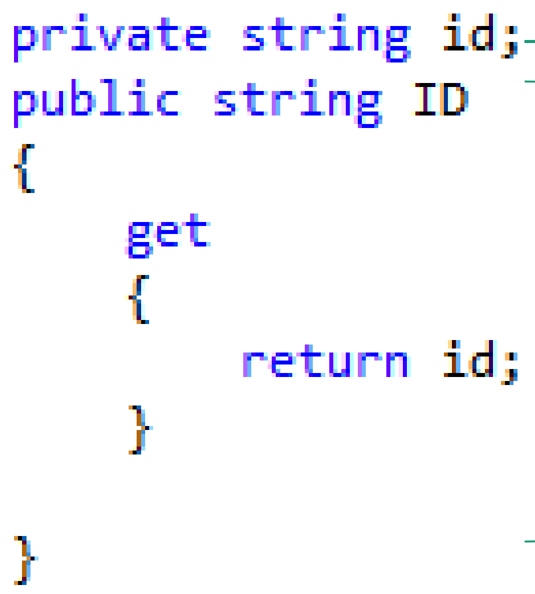
- Encapsulation is a technique used to protect the information in an object from another object.
- Hide the data for security such as making the variables private and expose the property to access the private data that will be public.
- So, when you access the property you can validate the data and set it.

Read-only Property

Read-only Property

- We can also create a read only property.
- Read only means that we can access the value of a property, but we can't assign a value to it.
- When a property does not have a set accessor then it is a read only property.

Read-only Property



```
private string id;
public string ID
{
    get
    {
        return id;
    }

}
```

Private Field id

Read-only Property

we can access the value of a property, but we can't assign a value to it

Abstraction VS Encapsulation

Abstraction: Hiding unimportant data, revealing only the important one

Encapsulation: Hiding the data for the the purpose of protecting data

References

- Object Oriented Programing , Virtual University , Lecture 2, Online Available at: https://ocw.vu.edu.pk/CourseDetails.aspx?cat=Computer+Science%2FInformation+Technology+&course=CS304
- Object Oriented Programing , Virtual University , Lecture 9, Online Available at: https://ocw.vu.edu.pk/CourseDetails.aspx?cat=Computer+Science%2FInformation+Technology+&course=CS304

THANKS