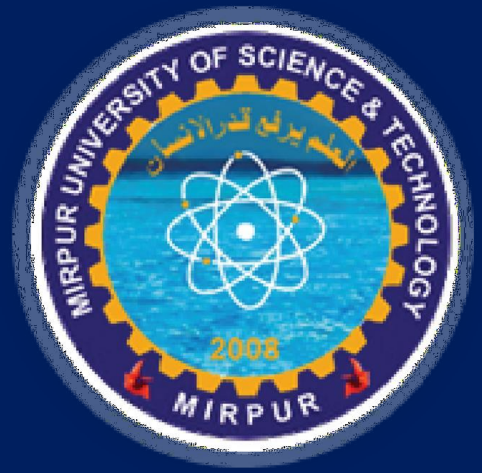


MUST

Wisdom & Virtue

MIRPUR UNIVERSITY OF SCIENCE AND TECHNOLOGY

DEPARTMENT OF SOFTWARE ENGINEERING

Object Oriented Programming

Lecture 4 : Introduction to a Class in OOP

Engr. Saman Fatima
Lecturer

Last Lecture

- **What is an Object?**
- **How to identify Objects?**
- **Examples**

This Lecture

- **What is a Class?**
- **How to create and use a class in C#?**
- **How to declare multiple instance of a class**

What is a Class?

- Class is a tool to realize objects in Object Oriented Programming Language (C#)

Object in OO Model

Student
Attributes: Name RollNumber Class Age
Behavior: Introduce
Identity: RollNumber

What is a Class?

- Class is a tool to realize objects in Object Oriented Programming Language (C#)

Object in OO Model

Student
Attributes: Name RollNumber Class Age
Behavior: Introduce
Identity: RollNumber

Derived Class in C#

```
class Student
{




}
```

What is a Class?

- Class is a tool to realize objects in Object Oriented Programming Language (C#)

Object in OO Model

Student
Attributes: Name RollNumber Class Age
Behavior: Introduce
Identity: RollNumber

Derived Class in C#

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

}
```

Data Members

What is a Class?

- Class is a tool to realize objects in Object Oriented Programming Language (C#)

Object in OO Model

Student
Attributes: Name RollNumber Class Age
Behavior: Introduce
Identity: RollNumber

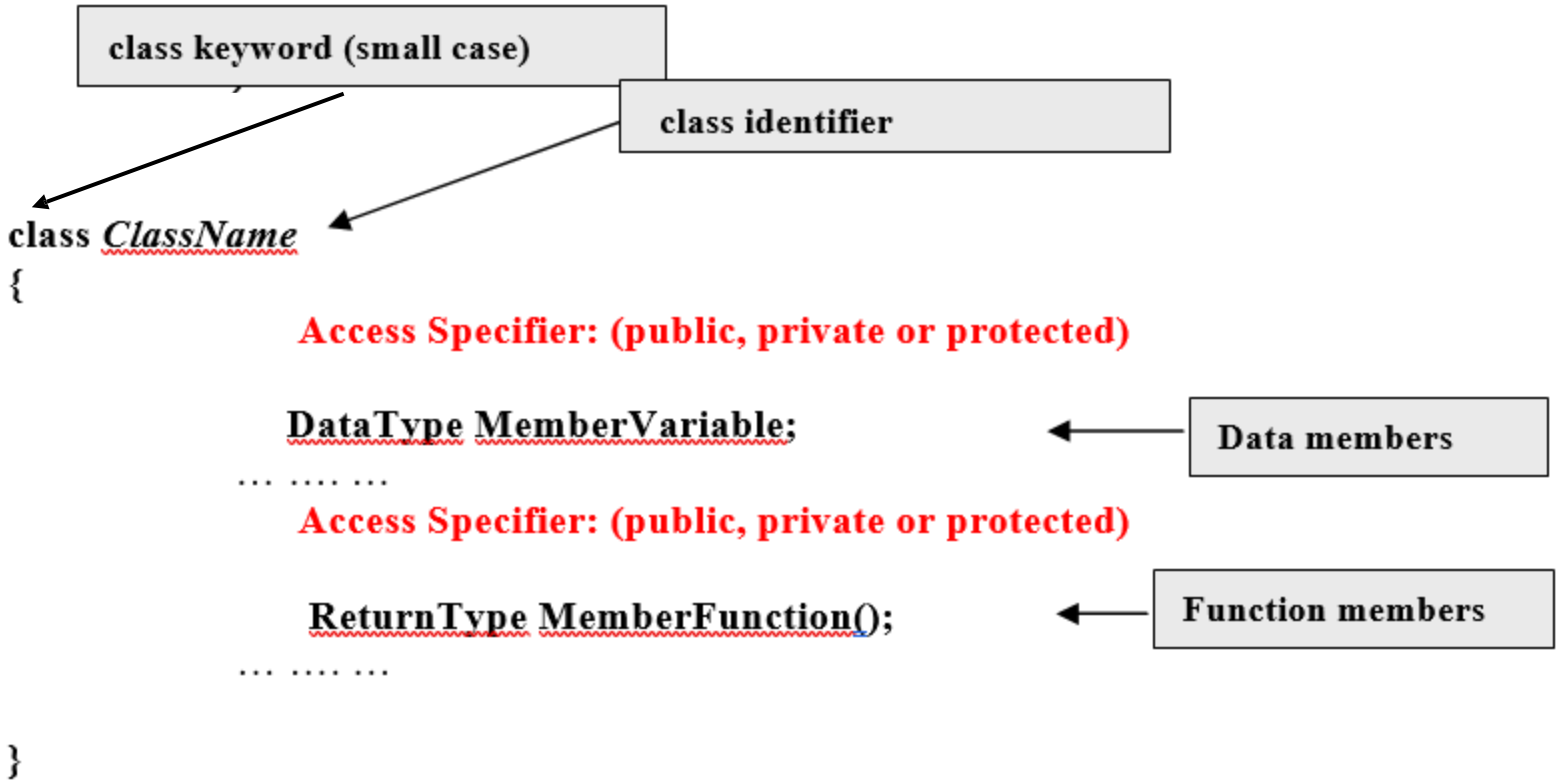
Derived Class in C#

```
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;
    }

}
```

Data Members

Member Function

Class Definition:



Object Use

After a class has been declared and defined, an object of that class can be declared (also known as creation or instantiation) and used, a class is just like another type (int, char, etc).

A programmer can declare an object with the following format:

ClassName ObjectName;

This statement creates an object based on the blueprint of class '**ClassName**' and the object can be referred to by the identifier (variable name) '**ObjectName**'

The ' . ' (dot) operator can be used to access an object's public members

The format for referring to an object's member is:

ObjectName.MemberFunction()

Use class to create object

```
Student student1 = new Student();
student1.Name = "Ali";
student1.RollNumber = 12;
student1.Class = "BSSE-1";
student1.Age = 18;
LblMessage.Text = student1.Introduce();
```

Multiple Instances of a Class

- In OOP we create a general sketch for each kind of objects and then we create different **instances** using this sketch
- we call this sketch or prototype or map as "**class**".
- All objects of same kind exhibit identical characteristics (information structure and behaviour) however they have data of their own.

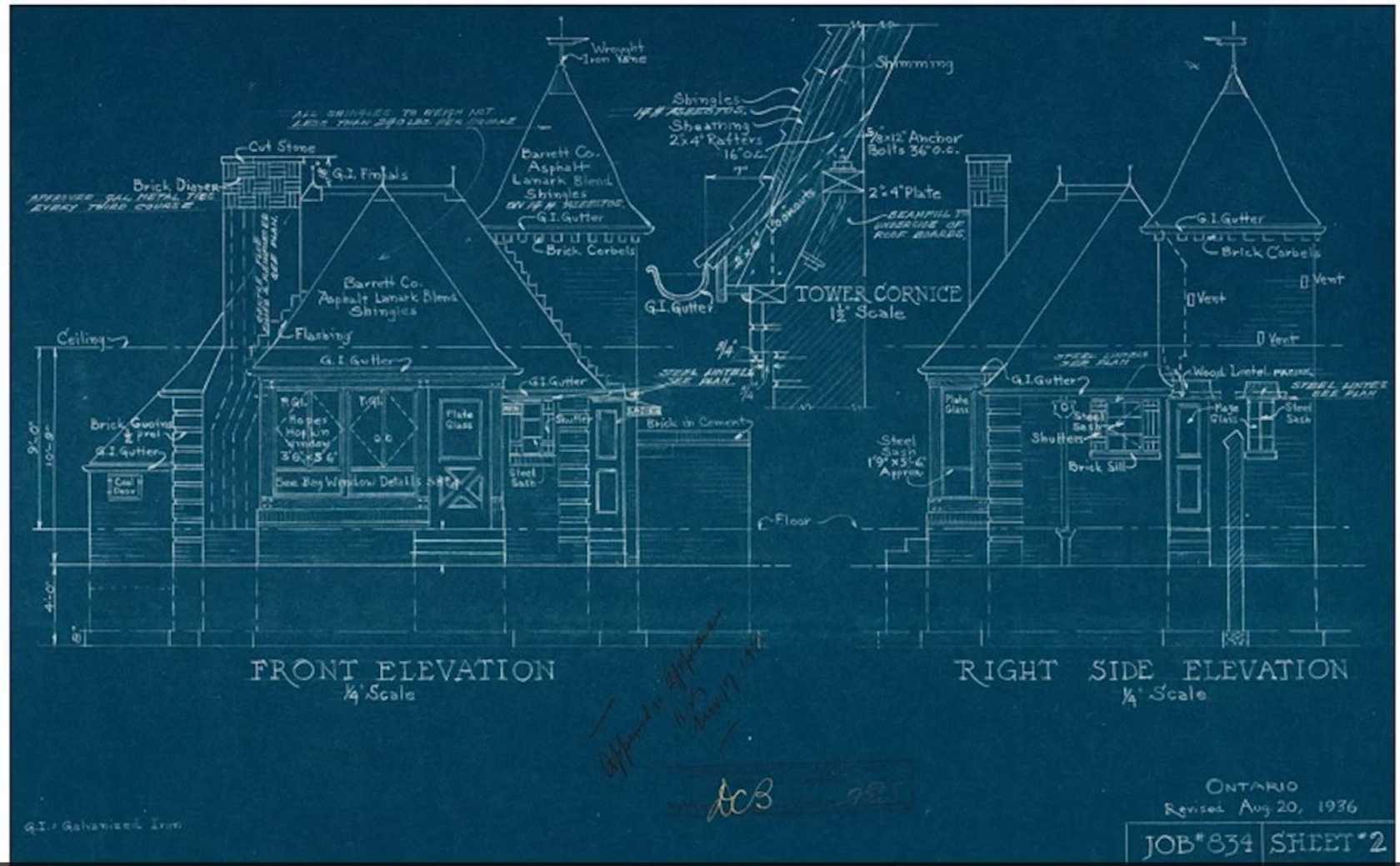
Code Demonstration

How to declare multiple instance of a class

Code Demonstration

In Object-Oriented Programming (OOP), you can declare multiple instances (objects) of a class by simply creating new objects from that class. Each object will have its own set of data, while all objects share the same structure (as defined in the class).

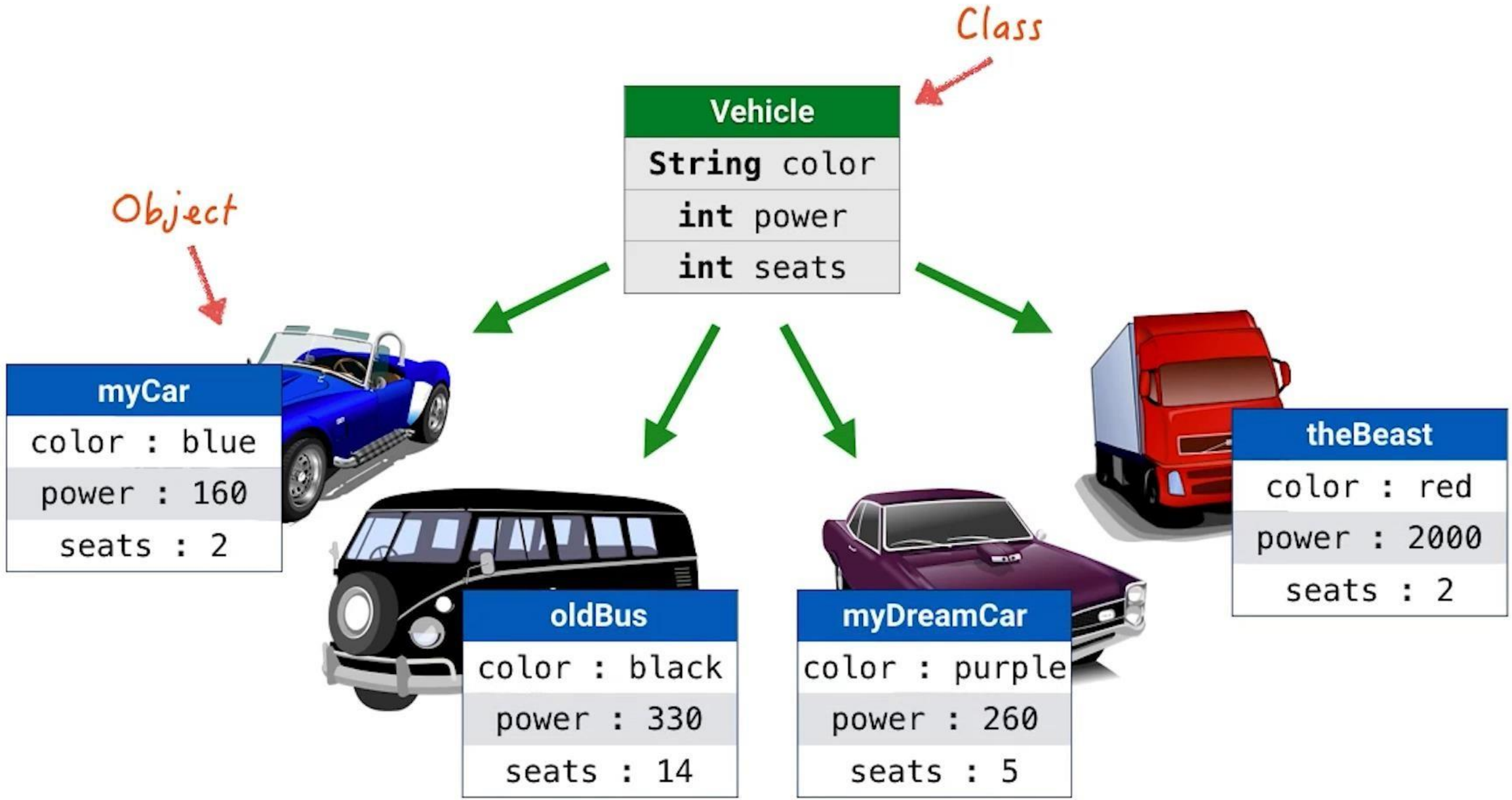
Classes & Objects



It's the blueprint that defines what the object should look like.

Classes & Objects

An object on the other hand is the actual entity that is created from that class.



Each object would have a set of different values for those same fields.

Code Example -1

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

Syntax for Declaring Multiple Instances

- // Declaring multiple instances (objects) of the class
- Student student1 = new Student();
- Student student2 = new Student();
- Student student3 = new Student();}

Syntax for Declaring class

```
public 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}";
    }
}
```

Syntax for Declaring class

```
public 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}";
    }
}
```

Example -1

```
class Program
{
   static void Main()
   {
      Student s1 = new Student() { Name = "Ali", RollNumber = 101, Class = "10th", Age = 16 };
      Student s2 = new Student() { Name = "Sara", RollNumber = 102, Class = "10th", Age = 15 };
      Student s3 = new Student() { Name = "Ahmed", RollNumber = 103, Class = "10th", Age = 17 };

      Console.WriteLine(s1.Introduce());
      Console.WriteLine();
      Console.WriteLine(s2.Introduce());
      Console.WriteLine();
      Console.WriteLine(s3.Introduce());
   }
}
```

References

- Object Oriented Programing , Virtual University , Lecture 3, Online Available at: https://ocw.vu.edu.pk/CourseDetails.aspx?cat=Computer+Science%2FInformation+Technology+&course=CS304

THANKS