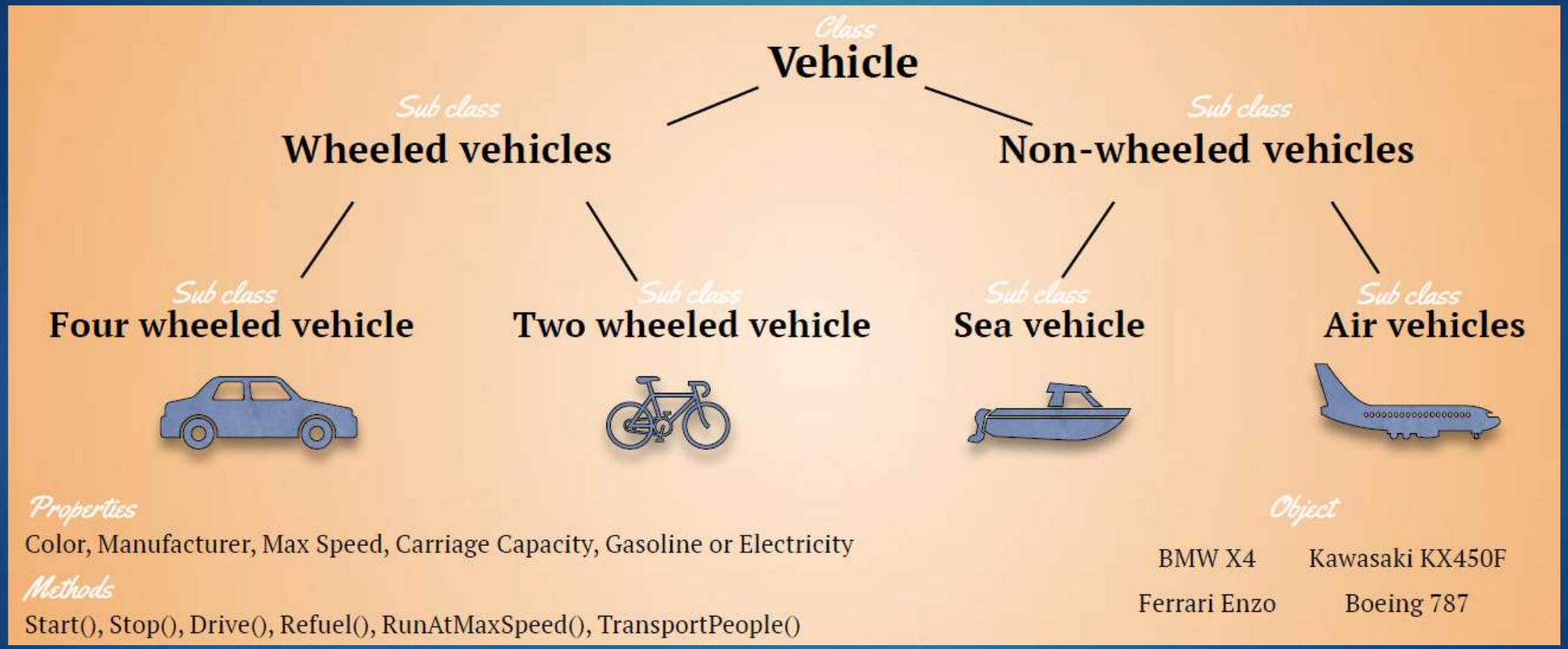
Lecture 3

CONCEPTS OF OBJECT-ORIENTED PROGRAMMING

Classes

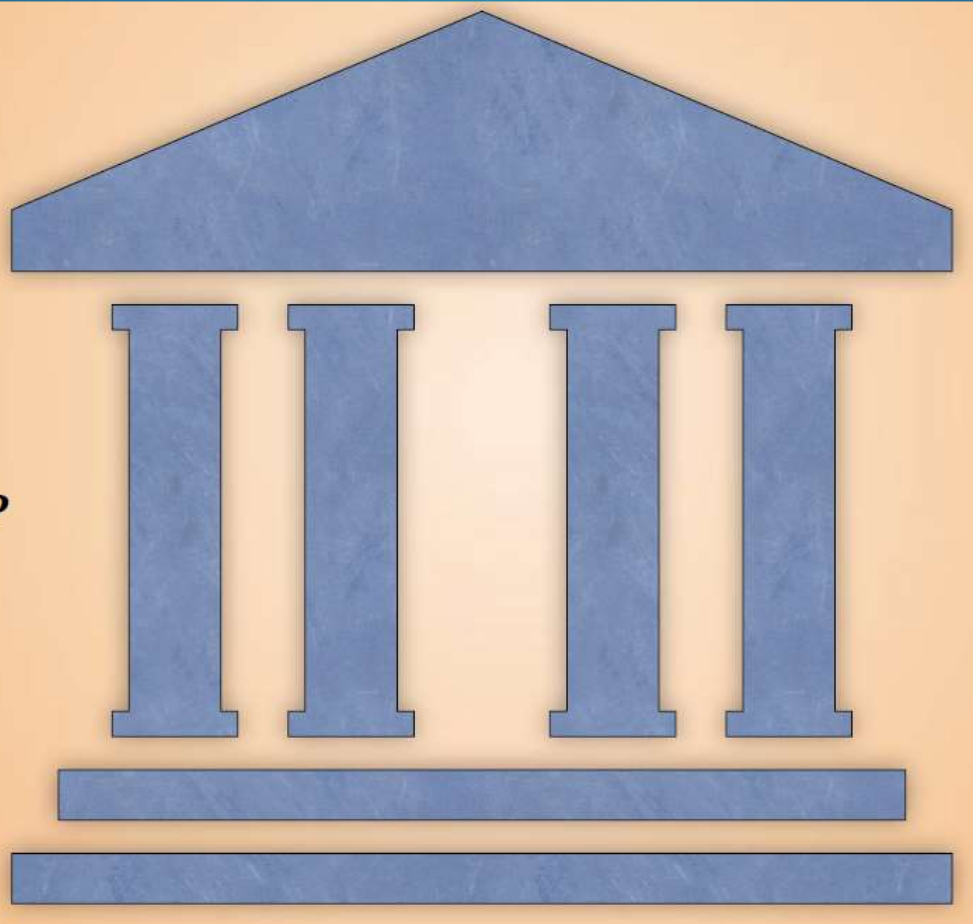
- ***Class : A category of objects. The class defines all the common properties of the different objects that belong to it.***
- ***Object : Refers to a particular instance of a class where the object can be a combination of variables, functions, and data structures.***
- ***Method : A combination of instructions grouped together to achieve some result. It may take arguments and return result.***
- ***Property : A member that provides a flexible mechanism to read, write, or compute the value of a private field***

What is OOP


Class
Vehicle
Sub class
Wheeled vehicles
Sub class
Non-wheeled vehicles
Sub class
Four wheeled vehicle
Sub class
Two wheeled vehicle
Sub class
Sea vehicle
Sub class
Air vehicles
Properties
Color, Manufacturer, Max Speed, Carriage Capacity, Gasoline or Electricity
Methods
Start(), Stop(), Drive(), Refuel(), RunAtMaxSpeed(), TransportPeople()
Object
BMW X4
Kawasaki KX450F
Ferrari Enzo
Boeing 787

What is OOP

The Four Pillars of OOP



Inheritance

Encapsulation

Polymorphism

Abstraction

What is OOP

- ***Inheritance : The process of creating the new class by extending the existing class or the process of inheriting the features of base class is called as inheritance.***
- ***Polymorphism : Poly means many and Morph means forms. Polymorphism is the process in which an object or function take different forms.***
- ***Abstraction : Abstraction is the process of showing only essential features of an object to the outside world and hide the other irrelevant information.***
- ***Encapsulation : Encapsulation is a process of binding data members (variables, properties) and methods together.***

Methods

- **A method is a code block that contains a series of statements. A program causes the statements to be executed by calling the method and specifying any required method arguments.**
- **Method name and its parameters types (but not the parameter *names*) are part of the signature.**

```
static void Main(string[] args)
    {
        int a, b, c, d;
        int addResult = 0;

        a = 5;
        b = 3;
        addResult = a + b;
        Console.WriteLine($" {a} + {b} = {addResult}");
        DisplayMessage();

        c = 15;
        d = 10;
        addResult = c + d;
        Console.WriteLine($" {c} + {d} = {addResult}");
        DisplayMessage();

    }
```

```
static void DisplayMessage()
    {
        Console.WriteLine("Process is done");
        Console.WriteLine("This process is run by
ahmad");
        Console.WriteLine("Finished on time :
"+DateTime.Now.ToShortTimeString());
    }
```

Methods

```
static void Main(string[] args)
        {
            int a, b, c, d;
            int addResult = 0;

            a = 5;
            b = 3;
            addResult =
PerformAddOperation(a, b);
            Console.WriteLine($" {a} +
{b} = {addResult}");
            DisplayMessage();

            c = 15;
            d = 10;
            addResult =
PerformAddOperation(c, d);
            Console.WriteLine($" {c} +
{d} = {addResult}");
            DisplayMessage();

        }
```

```
static int PerformAddOperation(int x, int y)
        {
            int addResult = 0;
            addResult = x + y;
            return addResult;
        }
```

```
 static void DisplayMessage()
        {

Console.WriteLine("Process is
done");
            Console.WriteLine("This
process is run by ahmad");

Console.WriteLine("Finished on
time : " +
DateTime.Now.ToShortTimeStrin
g());
        }
```

Methods

- **Passing by value (using a copy)**
- **Passing by reference (using the variable itself)**
- **ref keyword**
- **out keyword**

```
static void Main(string[] args)
      {
          //string firstEmployee, secondEmployee;

          //firstEmployee = "David Smith";
          //secondEmployee = "Sophia Watson";

          //Console.WriteLine($"Inside Main Method\n--
--------\n{firstEmployee} \n{secondEmployee}\n\n");

          ChangeNames(out string firstEmployee, out
string secondEmployee);

          Console.WriteLine($"Inside Main Method\n-----
-----\n{firstEmployee} \n{secondEmployee}\n\n");
      }
```

```
static void ChangeNames(out string firstEmp,out
string secEmp)
      {

          firstEmp = "Olivia Aaron";
          secEmp = "Alvaro Salazar";
          Console.WriteLine($"Outside Main Method\n-
---------\n{firstEmp} \n{secEmp}\n\n");
      }
```

Overloaded methods

```
static void Main(string[] args)
    {
        string guestName = "";

        Console.WriteLine("Hello, Dear Guest, what is your name?");

        guestName = Console.ReadLine();

        if (guestName == string.Empty)
            WelcomeGuest();
        else
            WelcomeGuest(guestName);

    }
```

```
static void WelcomeGuest()
    {
        Console.WriteLine("Okay, we hope you enjoy staying at our hotel");
    }

    static void WelcomeGuest(string name)
    {
        Console.WriteLine($"Thank you {name}, we hope you enjoy staying at our hotel");
    }
```