C# for Beginners

Encapsulation and data hiding

Agenda:

- Classes and objects
- Encapsulation
- Data hiding
- Access modifiers
- Properties

Farid Naisan, farid.naisan@mau.se, University Lecturer, Malmö University

Classes and objects

- A program in C# consists mostly of classes. Most programs are made of many classes, which can be tens, hundreds or thousands.
- A class is a collection of data and operations, or in programming terms, fields and methods.
- A class defines a group of similar objects. It is a blueprint from which one or more objects of that type may be created.
- An object created from a class is called an instance of the class. "Instance" and "object" are two words for the same thing and are used interchangeably.
- While a class is only a definition, an object is a universally unique “thing” that exists.
- Virtual objects get a unique ID and they are placed in the computer's memory.
  - The class Car describes cars in general.
  - `yourCar, myCar` are two unique objects of the class `Car`.

Class and object

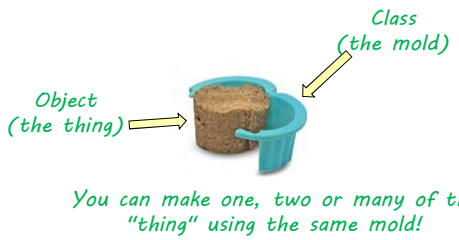
- Classes can represent physical objects like `Car`, `House`, `TV`, or conceptual objects like `Address`, `BankLoan`, or `Rectangle`.
- A class is a type while an object in the memory that is created from the class and is referenced by a reference variable.
- Objects are created by the keyword new and placed on the heap memory.
- A reference variable contains the address of an object.

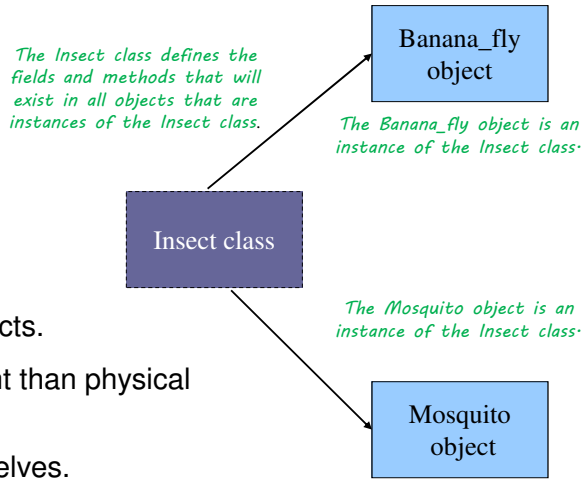
```
BankLoan loan = new BankLoan();
```

- It is very common to use the word "object" to a reference variable, which actually only has the address of the memory where the object is placed.
- Every instance of a class will have its own set of the values stored in the fields of the object.
- Every instance of a class will have a copy of the methods of the class.

Farid Naisan, farid.naisan@mau.se

Classes and Objects



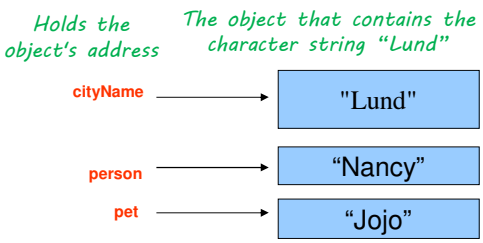


- Class is like a mold used to create objects.
- Virtual objects are much more intelligent than physical objects.
- They can perform operations on themselves.

Farid Naisan, farid.naisan@mau.se

Classes

- Many objects can be created from a class. Each object can be accessed (referred to) using a reference variable.
- A reference variable contains the address of an object in the memory.




```
string cityName = "Lund";
string person ="Nancy";
string pet = "Jojo";
```

- Class objects normally have properties and methods that perform useful operations on their data.
- The Length property of the string class returns an integer value that is equal to the length of the string.

```
int stringLength = cityName.Length;
```


Farid Naisan, farid.naisan@mau.se

Class definition

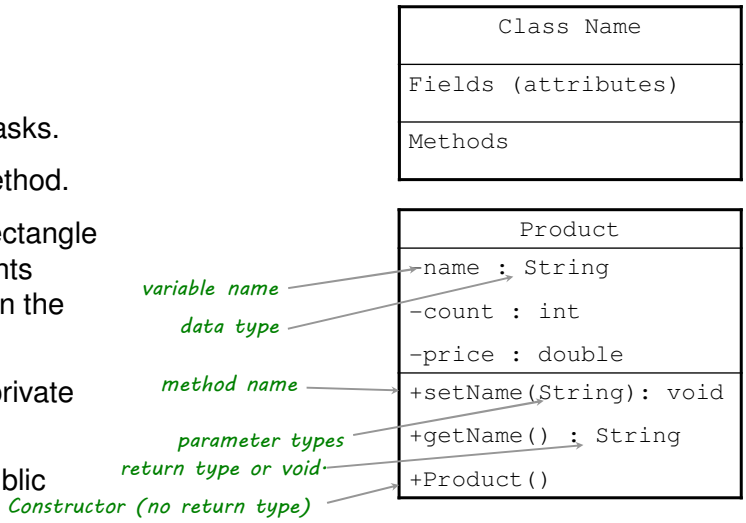
- A class is defined using the keyword class followed by a class name that must abide with the identifier naming rules.
- A general layout of a class may look as in the figure.
- At the top of the file, namespaces containing classes we need to use are included.
- A namespace is like a package name for the collection of types included in the in the package.
- Namespaces allow you to write classes with the same name but in different namespaces.
- Whenever the compiler does not recognize a class, it can happen that the namespace under which the class is located is not imported.
- It is always a good idea to use a namespace for every application.

```
3   using System.Collections.Generic;
4   using System.Linq;
5   using System.Text;

7   namespace ProductTest
8   {
9       public class Product
10      {
11          Fields

18

19          Constructors

40

41          Properties

74

75          Methods

108

109     }
110 }
```


Farid Naisan, farid.naisan@mau.se

Parts of a class

- A class contains:
  - fields for storing data,
  - methods for performing tasks.
- A constructor is a special method.
- Classes are modeled as a rectangle containing three compartments stacked vertically as shown in the figure.
- A minus sign (-) is used for private members.
- A plus sign (+) is used for public members.



Farid Naisan, farid.naisan@mau.se

Create a class

- Each of the objects (dishWasher, coffeeMaker, egg and milk), has its own set of values stored in the fields that are declared in the class.
- The values describe the status of the object at a certain time.
- Each of the objects have a copy of the methods, like **Price** that can be called using the object name.
- Price is a Property, a special method that is a convenient way of accessing a private field, replacing setter and getter methods in other languages.

```
public class Main
{
    public static void main(String[] args)
    {
        Product diskWasher = new Product();
        Product coffeeMaker = new Product();
        Product egg = new Product();
        Product milk = new Product();

        diskWasher.Price =5000.0;
        egg.Price = 44.0;
    }
}
```

```
public void setPrice(double price)
{
    if (price >= 0.0)
        this.price = price;
}
```

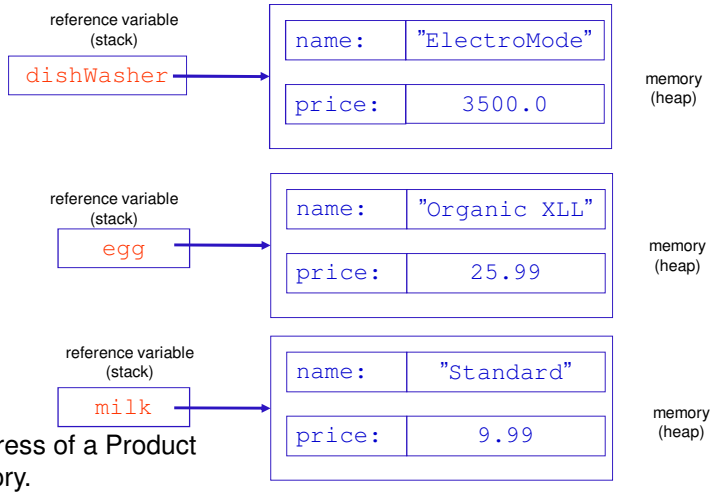

Farid Naisan, farid.naisan@mau.se

Status of the three different Product objects

The variable **dishwasher** is a reference variable and has the address of a Product object in the computer's memory. The object it self with its data is saved in that address. To access the values, we use dot-notation. dishwasher.name.

The variable **egg** has the address of a Product object in the computer's memory.

The variable **milk** has the address of a Product object in the computer's memory.



Farid Naisan, farid.naisan@mau.se

Fields

- Fields, instance variables, member variables or attributes are different names for the same thing, but Fields is the "official" word and is used more often.
- Fields should include attributes that describe the objects' status. They should be properties that make parts of the class.
  - Color, horse power, model are examples of fields that describe a Car object, not the owner which is not a part of the object.
  - An Owner is another type of a class but it can use one or more objects of the Car class.
  - Every object of the Car class will have its own set of fields, as every object is a unique specimen of the Car class.
- Do not use fields for saving temporary data. Use local variables (variables inside a method) instead.

Farid Naisan, farid.naisan@mau.se