MALMÖ HÖGSKOLA

Programming Using C#,
Basic Course

Object-Oriented Programming with C#

Agenda:

- Introduction to basic Object-Oriented Analysis and Design (OOAD) concepts
- Advantages of object-orientation
- Objected oriented programming (OOP)
- Classes and objects

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

OOP languages

- The term Object Oriented Programming is usually referred by its abbreviation OOP.
- C++ is not a pure OOP language, but it gives full support for it.
- C#, Visual Basic and Java are pure OOP languages.
- Questions:
  - What does the term “OOP Language” mean to you?
  - Does the choice of an “OOP-language” make a difference in writing OO-code?

What is OOP?

- OOP is, in fact, an ideology, a paradigm!
- OOP languages offer only support for implementing the principles of Object-Orientation.
- Some languages like C#, Java, C++, SmallTalk make OO thinking easier.
- Some languages like Pascal and Basic make it harder.

OOP languages offer facilities

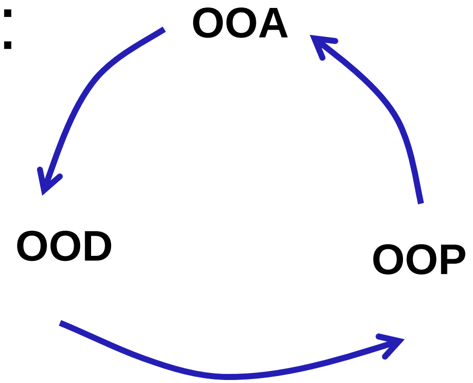
- Some languages enforces rules of OOP.
  - A C# or Java application consists only of classes.
- Others prohibits non-OOP techniques.
  - C# does not allow global variables.
- To take better advantage of an OOP-language, it is necessary to have a good understanding of:
  - the subject Object-Oriented Analysis and Design (OOAD) in general and
  - the Object-Oriented Programming (OOP, a part of OOAD) in particular.

OOP comes by experience

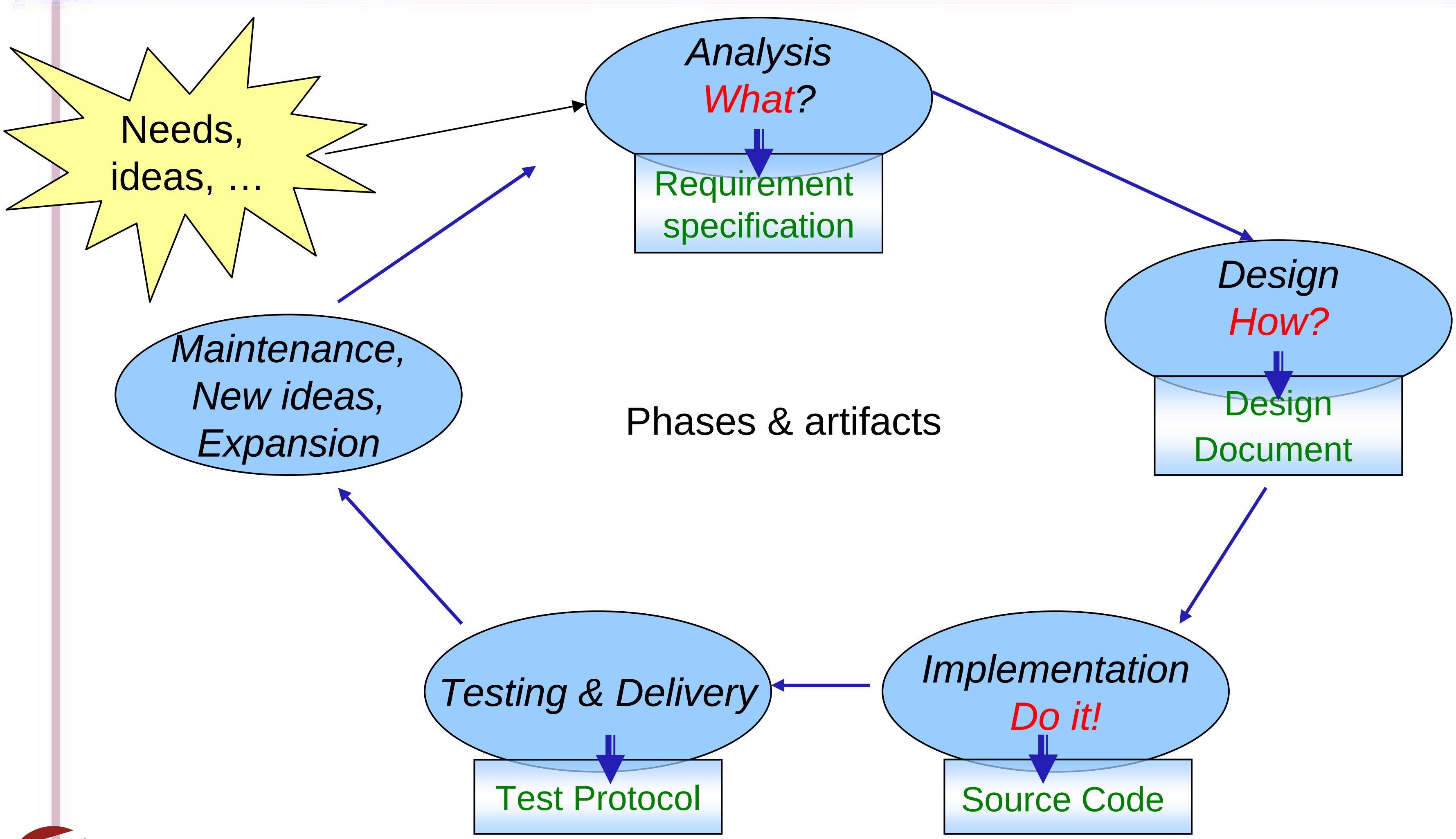
- How object-oriented an application development is, depends highly on the programmers' knowledge and experience of the theory and understanding of the OOP principles.
- One can be less object-oriented in a pure OOP language like C# and Java.
- One can be very object-oriented in a non- OOP language like C.

OOP is a part in Object-Oriented Analysis and Design - OOAD

- Software development is nowadays an iterative process. OOAD contains three consecutive stages:
  - OOA: Object-Oriented Analysis: *What* to do?
  - OOD: Object-Oriented Design: *How* to do?
  - OOP: Object-Oriented Programming: *Do* it!



A general view of the software development process



Object-orientation and programming

- Object-orientation is to think about objects that an application or a software system needs and how they should be related.
- OOP, Object-Oriented Programming is usually not the biggest part of OOAD.
- A good OOA and OOD makes OOP much easier and efficient.

OO a model of reality

- The main goal of OO is to solve complexities.
- OO works with “things”, objects just as in real world.
- Effort is made to apply a model of real objects.
- An application is divided into smaller parts (objects).
  - Every part is a special “thing”.
- Begin to make generalizations – abstractions.

Differentiate between class and object

- The terms “class” and “object” are used very interchangeably, often as a matter of carelessness.
- An object is a “thing”, and a thing in object-oriented terms can be a real object, virtual or a a conceptual thing,
  - person, a car, an address, a solution to a mathematical equation are some examples.

Objects

- Every objects in terms of OO is responsible for its state (properties) and operations (methods).
- Just as every thing in the world is a unique thing, an object is very unique.
  - Your car and your friends car have same model, manufacturer, color and all other attributes; they are similar (belong to same class) but universally two specific and very unique objects (instances of the class).

Class

- Similar attributes and behavior of the objects make a class.
- A class defines a group of similar objects.
- A Car class can be written to define cars from some level of abstraction, from a particular perspective.
- Depending on the system that uses the object, a car class can be defined in different ways.

Data and operations

- For a car seller, the color, model, price are some of the data of interest.
- For a registration office, the chassis number, weight, and model of a car are interesting data.
- Objects in OO also perform tasks.
    - Start, Stop, Accelerate are some of operations that can be expected from a car object.

Parts of a class

- Objects have data and operations.
- The data is called attributes, or properties or simply data.
- The operations are called behaviors, or services, or operations or responsibilities.
- A car object can have attributes:
  - “BMW 1995”, 140 hp, Green
- It can do:
  - Accelerate to reach x speed in s seconds.
  - It can stop to zero speed in some units of time.

Class data and operations

- A class is a form for making objects.
- A simple car class can look like this:

```
class Car
{
  string model;
  double hp;
  Color  color;

  void   AccelerateToReach(double speed)
  {
   //statements
  }
  void   Stop()
  {
   //statements
  }
}
```

Attributes and methods

- The data is usually called as Fields, Attributes or Variables of a class.
  - attributes define the state of the object
- The operations are usually called Methods or simply functions of a class.
  - Methods perform tasks using the data.
  - It offers services to other objects.

Objects in OOP are smart

- From an OOP perspective, an object should be much smarter than a real world’s object.
- A blackboard object should be able to, for example, erase itself, undue erasing, open itself, close itself and save its data.
- An application usually uses more than one object.
    - The objects communicate through messages to each other.

Advantages of OO

- Better communication
  - The OO approach makes it easier for others to get easily involved in the software development and maintenance.
  - Using the OO principles, a unified modeling language (UML) and same methodology provides a good means of communication.
- Code Re-use
  - Apply the OOP technology, the code can be re-used in the application as well as in different projects.
  - Objects developed by other parties can be easily incorporated into the system.

Advantages

- Model of real-world objects
  - The closer relationship between the objects in the software and the objects of the real-world results in a good software product.
- Preventing surprises
  - The models gives an early picture of what is to be expected from the system. Problems and complexities are discovered in an early stage.

Advantages

- Better Budgeting
  - The OO approach gives a good idea of how large and complex a system can get. It makes the estimation of the costs and resources easier.
- Team-work gets easier
  - Developer groups can get work on different segments of work. They can develop and test the objects independently before putting them in the system.

Advantages

- Better planning and better progress measuring
  - The different phases makes usage of milestones a lot easier.
  - A good scheduling makes a good follow-up also possible.
- Effective Maintenance Management
  - Problems, bugs and shortages of the software can be located a lot easier.
  - Expanding the software without rewriting is no problem in OOP.

The three important OOP principles

- OOP is the art of implementing the principles of the object-oriented technology.
- OOP does not really have a standard definition.
- It has some key concepts. The most important ones are:
  - Encapsulation
  - Inheritance
  - Polymorphism
- Other concepts such as abstraction, data hiding and dynamic binding may also be added to the list, but they may also be considered as parts of the above.

Defining objects and their members

- To define objects and determine the relation between them are the most important and hardest part of the design.
- Once objects are defined, classes are to be designed.
- To define an appropriate set of data and methods of a class, it is important to find out how an object of the class is used.

Encapsulation

- Encapsulation is one of the basic aspect of OOP.
- It means to bundle data and operations that are related to each other in an entity, namely a class.
- This principle has many advantages.
  - It reduces human errors as the functions can check the validity of data.
  - Data hiding can be applied in the entity to protect data and the internal implementations.

Data hiding

- A lot of people think of data hiding as a part of encapsulation.
- Others distinguish data hiding from encapsulation.
- No matter which definition is used, it is more important to encapsulate properties and methods related to each other in the same class and to hide the private parts.

Hide the private parts of a class

- Protect the data by declaring the attributes private.
- Hide the internal behavior of the object by making internal operations private.
- Access to data is to be managed through special public methods, the so called setter and getter methods.
- Services are offered by some public methods.

Inheritance, Extension and Specilization

- Inheritance means to extend an class’s data and functionality.
- A sub-class inherits a base-class without rewriting the code. It extends another class, the base class.
- Inheritance enables reuse of code.
- A more specialized class, a sub-class, is derived from another class, a base-class (or super-class).

Polymorphism

- Polymorphism is to make different objects work in the same way.
  - All tape-recorders have a play-button, but how they function internally depends on the manufacturers know-how – and it is of no interest to the user!
- Polymorphism is achieved in different ways, for example by inheritance (virtual methods), abstract classes, dynamic-binding and interfaces.

Interface

- Interface has a special meaning in OOP terminology.
- An interface is like a class but without implementation.
- An interface can be implemented by other classes.
- It defines properties and methods that must be implemented wholly when a class implements the interface.
  - It is like signing a contract.

Summary

- Object-oriented analysis, design and programming are related to each other but distinct from each other.
- OOA, object-oriented analysis results in a object model of the application domain. The result are used in determining requirements.
- OOD, object-oriented design results in a object-oriented system model. The model is used to implement the design of the requirements.
- OOP, object-oriented programming is the programming part. It realizes the OOD model using an OOP language like Java, C#, and C++.