Assignment 1, (week 35)

In this weeks' assignment you will demonstrate your skills regarding inheritance and polymorphism – two key concepts in object-oriented programming. You will demonstrate this theoretically, and you will demonstrate that you know how to implement these concepts and techniques in code. This is the beginning of your journey where the end game is to develop well-structured, easily understood, maintainable, extendable, and robust software.

Hand-in instructions and requirements

Reading requirements and instructions is an important part of software development. Always make sure that you clearly understand the conditions to pass an assignment or performing a coding job. It's better to ask a question rather than doing the work twice!

Hand in a report with course code, course name and your name to the course room in Learn. The report should also consist of your answers to the assignment and questions in accordance with their respective descriptions.

Hand in a written report that contains:

- Your data model of an inheritance relationship (1)
- Answers to the questions (2). You must include the corresponding number for the question, for instance 2.1 answer, 2.2. answer and so on.
- Optional: write a short reflection on what you think about this assignment. What was hard, what was fun etc. (optional).

Programming assignment

For the programming assignment (3) you must hand in your .sln file along with your source code files and project folders. All files must be zipped into one folder that you hand in on Learn.

Requirements for the programming assignment:

- Visual Studio Enterprise Edition (see *course room in Learn → course material → 02. Essential hints that will ease your work in the course* for license information and how to download). This will be required later in the course, so you might as well install the proper version right away.
- Your project must be .NET 5 – if you use a previous version, you will automatically fail.
- Your code should be clean, easy to read, and your code should follow .NET naming and indentation convention. If you can't present your code in such a manner you will automatically fail.
- Use the debugging tool in visual studio – this is crucial for aspiring testers!

Both the report and the zipped coding project must be uploaded to the course room in Learn.

Decide on one language for the report. And decide on one language for programming. It's ok to write the report in Swedish but to do the coding in English and vice versa, but do not mix languages within the report or within the project.

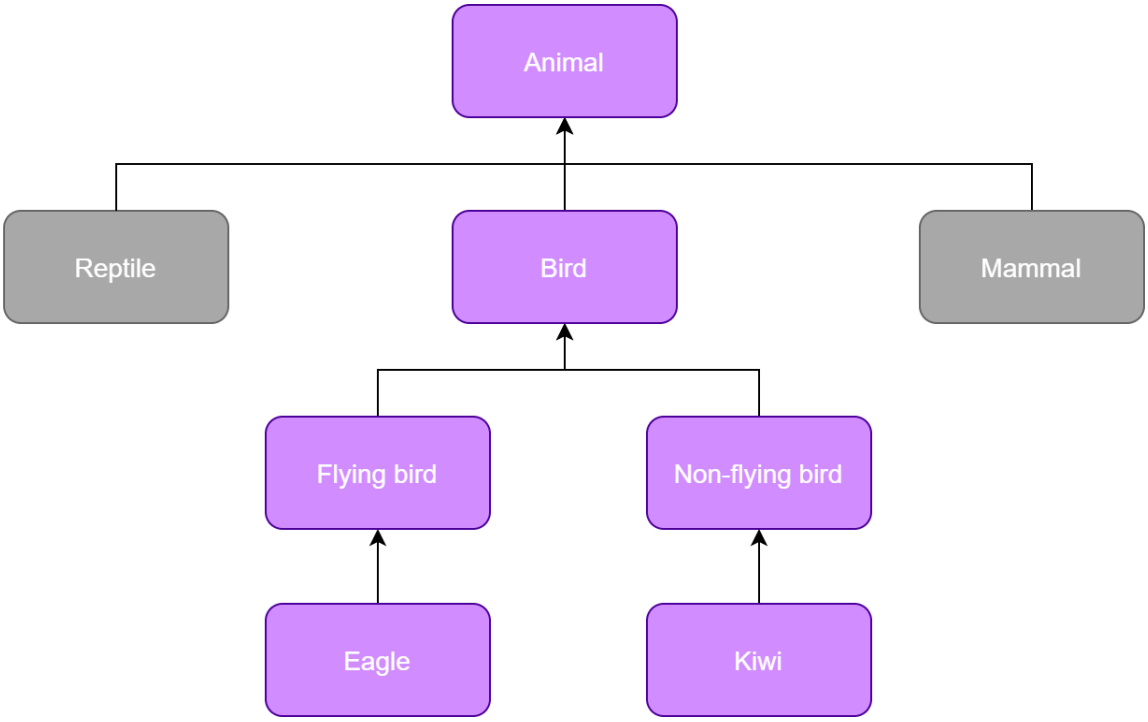
Deadline: 2021-09-07 at 17:00.

1. Create a data model of an inheritance relationship

The results of this part of the assignment must be presented in the report.

Example

The model below is modelling an inheritance hierarchy of four layers. First, we have the base class Animal on top. Second, we have three classes inheriting from the Animal class: Reptile, Bird and Mammal class. Third, we have two classes deriving from the Bird class, there are the Flying Bird class and the Non-Flying Bird class. Finally, we also have an inheritance relationship between the Non-Flying Bird class and the Kiwi Class where the latter inherits the former. Similarly on the left-hand side below the Bird class we have the Flying Bird class which has a hierarchical inheritance relationship with the Eagle class where the Eagle class derives from the Flying Bird class.



The assignment

Model an inheritance hierarchy of your choice in **four levels** of inheritance with a base class on top. The model should demonstrate a hierarchical inheritance relationship model similar to:

Animal(1) ← Bird(2) ← Non-Flying Bird(3) ← Kiwi(4).

The hierarchy should consist of four layers. If you cannot think of anything you can continue to create four subclasses of the class Mammal or Reptile, but please note that you must go two layers further since you're given the base class in this case.

You can use Draw io, Lucidchart, Visio or any other modelling tool you wish. Hand in your model in your report.

2. Answer the following questions

The results of this part of the assignment must be presented in the report.

Answer the following questions that all are related to inheritance or polymorphism. Short answers are perfectly fine. Hand in you answers with their corresponding number (for instance 2.1 answer, 2.2 answer) in your report. If you do not number your questions, you will not pass.

2.1. What is the etymology for the word class in an object-oriented programming context?

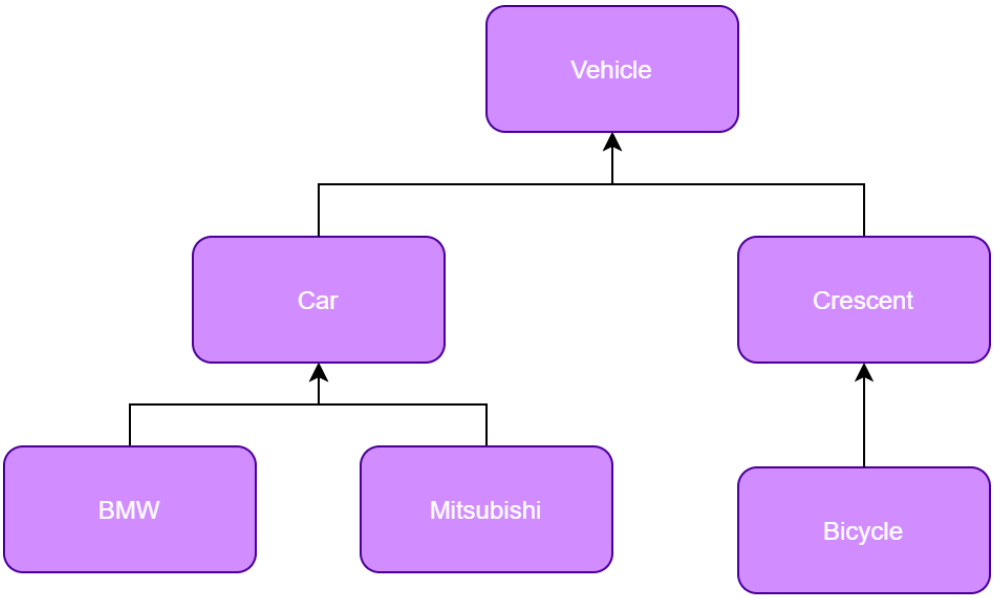
2.2. What is a base class?

2.3. What is a sub class?

2.4. How does inheritance support the reuse of software?

2.5. How does inheritance support polymorphism?

2.6. Is this correct, from a hierarchical perspective? (See model below). Motivate your answer; you should answer why or why not.



2.7. Is-a and has-a are examples of relationships. Which relationship should every derived class have to its' base class?

2.8. C# supports single-inheritance. What is the difference between single-inheritance and multiple-inheritance?

2.9. Explain the purpose of polymorphism.

2.10. What is the difference between overriding a method and overloading a method?

2.11. How does method overriding relate to inheritance?

2.12. What are the types of polymorphism in C# and how do they differ?

2.13. When should you use polymorphism?

3. Programming assignment

The results of this part of the assignment must be included in the zip file.

Create a base class that manages different shapes, the class Shape. The class must have two methods. One that calculates the area and one that calculates the circumference.

When you are done with the base class, create at least three child classes (Circle, Triangle and Rectangle) that inherits from the base class Shape. The derived classes should overload the methods from the Shape class. If you want to, you can create additional methods and properties that can be of interest for the different shapes.