

# Exercises

---

October 3rd 2017

This exercise sheet can be downloaded as [PDF](#)

## Exercise 1 - Create an extension method

Extend the functionality of `IList<T>` to return the middle element of the collection.

This will involve creating a static class, that extends the functionality. An example can be seen below:

```
public static T GetMiddleElement<T>(this IList<T> collection)
```

## Exercise 2 - Create GradedCourse, BooleanCourse and Project classes

A `BooleanCourse` encapsulates a course name and a registration of passed/not passed for our sample student.

A `GradedCourse` encapsulates a course name and the grade of the student. For grading we use the Danish 7-step, numerical grades 12, 10, 7, 4, 2, 0 and -3. The grade 2 is the lowest passing grade.

In both `BooleanCourse` and `GradedCourse` you should write a method called `Passed`. The method is supposed to return whether our sample student passes the course.

The class `Project` aggregates two boolean courses and two graded courses. You can assume that a project is passed if at least three out of the four courses are passed. Write a method `Passed` in class `Project` which implements this passing policy.

Make a project with four courses, and try out your solution.

## Exercise 3 - Rewrite GradedCourse and BooleanCourse to use inheritance

Create a base class called `Course` which `BooleanCourse` and `GradedCourse` inherits from.

Move common fields to the `Course` class.

Change the `Project` class to accept `Course` classes and adapt the methods `Passed()`.

If you can see a better way to use inheritance here, implement it that way. You decide between viable solutions.

```
public class GradedCourse : Course {  
    ...  
}
```

Try out your solution.

