

IT UNIVERSITY OF CPH

---

## BDSA AS01

---

Jens Birk Andersen - [jbia@itu.dk](mailto:jbia@itu.dk)

Thea Ullegård Kjeldsmark - [tkje@itu.dk](mailto:tkje@itu.dk)

Markus Petersen - [mgrp@itu.dk](mailto:mgrp@itu.dk)

BDSA  
IT-UNIVERSITETET I KØBENHAVN

September 13, 2021

# Contents

<b>1</b>	<b>C#</b>	<b>1</b>
<b>2</b>	<b>Software Engineering</b>	<b>1</b>

# 1 C#

## 1.1 Generics

**For the first method:** The type constraint means that T must implement the interface `IComparable`. Thus, elements of type T should be able to be sorted.

**For the second method:** The type constraint implies that T must extend U. At the same time, U should implement the interface `IComparable`, which means that elements of type U should be able to be sorted.

# 2 Software Engineering

## 2.1 Exercise 1

Knowledge acquisition can be generalized as non-linear or non-sequential because new pieces of information can potentially invalidate prior knowledge. As a result, several paradigm shifts have been seen in the history of humankind.

An example of non-sequential knowledge acquisition was the historical proposition that all life originated from an all powerful being i.e. "God". However, with new knowledge acquisition about natural selection, this idea was proven untrue or at least unscientific.

Another example where non-sequential knowledge acquisition was relevant is regarding the benefits of working from home. Before the Covid-19 pandemic, it was difficult for employees to work from home as businesses generally preferred physical attendance. However, due to the pandemic, the remote working options have drastically been improved as businesses gained knowledge about the beneficial aspects both for employees and the business in terms of availability of shifts and flexibility for employees.

Thus, if knowledge acquisition was sequential, we would be unable to change our minds and use further knowledge to improve our current ways of doing things.

## 2.2 Exercise 2

1. The first decision is based on system design.
2. The second decision would also be a system design decision if seen through the eyes of a software engineer.
3. The third decision is a decision regarding the requirements of the service.

## 2.3 Exercise 3

The first two uses of "account" in the paragraph are used as an application domain concept because the terms refer directly to the problem of accessing bank accounts within the functionality of the application; i.e within the scope of the "application domain".

The last two occurrences refer to the solution domain because they refer to the illusion of accessing bank accounts as a software solution for customers. They therefore provide a more abstract definition of the term "account" than would otherwise be considered in the application domain.

## **2.4 Exercise 4**

When constructing an aircraft or a bridge, it is possible to use data from previous similar projects to estimate both the required persons, time, and money. With the case of Word for Windows, these estimations were very difficult to make as the software development project was very complex and innovative.

Furthermore, it is generally easier for construction workers to coordinate the required tasks for completing a construction. For software developers, unexpected issues are much more frequent because software is invisible, which makes it more difficult to visualize compared to a real world construction.

Developing software, typically also requires a higher level of abstraction and it is often subject to changes. Thus, unforeseen complications can more easily occur than in constructing real world building projects.

## **2.5 Exercise 5**

Functional requirements determine the functionality of the application or its components. On the other hand, non-functional requirements, determine the performance standards, quality attributes, usability, security, scalability, etc.

The first statement is a functional requirement. The second one is neither as it is not pertinent to the functionality of the program nor the performance or quality of the program. The third one is a non-functional statement. The fourth is a non-functional statement. The fifth is a functional statement.

## **2.6 Exercise 6**

The purpose of creating models is to reduce complexity and increase comprehensibility by making a simplified version of the real world that can convey the same meaning but without unnecessary information.