



INF 101 Syllabus

Spring 2020

Course instructor: Martin Vatshelle

e-mail: martin.vatshelle@uib.no

Course objectives

The student will be able to:

- Use modern software development tools
- Implement abstract data types representing sound abstractions
- Use test driven development
- Evaluate design strategies: inheritance vs composition

Basic Information

Software:

In this course one of the main objectives is to learn to use modern software development tools. It is expected that after completing this course you are familiar with the following software:

- GIT
- Integrated development environment (IDE) – Eclipse, IntelliJ or similar
- Java 13
- Junit 5

Obligatory Assignments

There will be two obligatory assignments this semester, required to be handed in 6 March and 24 April.

Grading

The grading scale used is A to F. A is the highest passing grade in the grading scale, grade F is a fail. Each obligatory counts for 15% of your final grade, for a total of 30%. The final exam is worth 70% of your grade.

Important course links and books

- Mitt.uib course page: [INF101 20V / Object oriented programming](#)
- GitLab weekly labs: <https://retting.ii.uib.no/inf101.v20.oppgaver>
- Princeton online course textbook: <https://introcs.cs.princeton.edu/java/home/>
- Oracle Java tutorials: <https://docs.oracle.com/javase/tutorial/index.html>

Course literature

Introduction to programming in Java

Basic knowledge needed to get started with this course is covered in the first three chapters of *Introduction to Programming in Java*, published by Princeton. You can access reading and code materials from this site: <https://introcs.cs.princeton.edu/java/home/>.

The Java tutorials from Oracle

The advanced features of Java and the Object oriented programming concepts are covered in :

[The Java Tutorials from Oracle](#) – below details the tutorials covered in this course:

Getting started [[link](#)]

Learning the Java Language [[link](#)]

- Object-oriented Programming Concepts [[link](#)]
- Language basics [[link](#)]
- Classes and Objects [[link](#)]
 - Classes
 - Objects
 - More on Classes
- Interfaces and Inheritance [[link](#)]
- Numbers and Strings [[link](#)]
- Generics [[link](#)]
 - Why use generics?
 - Generic types
 - Generic methods
 - Generics, inheritance and subtypes

Essential Java Classes [[link](#)]

- Exceptions
- Basic I/O

Collections [[link](#)]

- Introduction
- Interfaces
- Implementations
- Algorithms
- Custom Implementations

Junit testing

DZone – Unit Testing [[link](#)]