

# FAGAFTALE: PROGRAMMERING 1

## (1. OG 2. SEMESTER F25)

### 1. SEMESTER

#### BESKRIVELSE

I programmeringsfaget på 1. semester lærer du at programmere i grundlæggende objektorienteret programering (OOP) med programmeringssproget Java. Du lærer også at anvende JetBrains IntelliJ, som er et udviklingsværktøj, der er udbredt i branchen. Du begynder også at anvende et professionelt versionsstyringssystem, som du kan bruge dels til backup og dels til at dele din kode med dine gruppekammerater i skyen.

Du lærer hvordan man øger programkvaliteten, dels ved at optimere koden systematisk og dels ved at følge nogle udbredte principper for ren kode i branchen, og hvordan man skriver automatiserede tests. Endelig lærer du forskellige måder at debugge sine programmer på for hurtigere at finde og rette fejl i koden.

#### EKSAMENSFORUDSÆTNINGER

- Deltagelse i tre gruppeprojekter med godkendt indsats.

#### LÆRINGSMÅL

- A. At kunne implementere klasser med konstruktør(er), attributter og metoder.
- B. At kunne forklare og anvende variabler, objekter, primitive datatyper, enums, metoder og kontrolstrukturer til selektion og iteration: if, switch, for, for each, while, do-while og ternary operator.
- C. At kunne anvende IntelliJ til udvikling af javaprogrammer.
- D. At kunne anvende basic kommandoer (init, add, commit, branch, switch) git til lokal versionsstyring.
- E. At kunne anvende GitHub til deling af kode med andre udviklere og håndtere merge konflikter.
- F. At kunne anvende JUnit til at implementere en simpel unit test efter Arrange/Act/Assert
- G. At kunne implementere fejlhåndtering med Exceptions (try- catch, throws) især i forbindelse med input fra brugere og filer.
- H. At kunne udarbejde et klassediagram, som ikke er autogenereret.
- I. At kunne deltage i kodereviews som giver og modtager.
- J. At kunne fejlfinde ved tracing (System.out.println) og brug af debuggeren.
- K. At kunne implementere en java applikation som løser en problemstilling.
- L. At kunne deltage i gruppearbejde og inddrage elementer fra semestrets fagområder (Programmering, Systemudvikling samt IT og forretningsudvikling) i løsning af en problemstilling.
- M. At kunne kritisk anvende internetressourcer udeover google og ChatGPT, herunder især stackoverflow og w3schools.
- N. Har kendskab til Java's officielle dokumentation (Java API docs)
- O. At kunne forklare og anvende objekt orienteret principper herunder encapsulation, arv, abstrakte klasser, interfaces og polymorfi.
- P. At kunne refactorere kode ud fra designprincipper
- Q. At kunne anvende datastrukturer array og ArrayList.
- R. At kunne implementere sortering i lister ved brug af Comparable og Comparator.
- S. At kunne persistere data i tekstdokumenter
- T. At kunne bruge Scanner til at modtage brugerinput og læse fra filer.

## LITTERATUR OG LÆREMIDLER

### Værktøjer

- IntelliJ Ultimate Edition: <https://www.jetbrains.com/idea/download> bruges som IDE hele semestret

## 2. SEMESTER

### BESKRIVELSE

Programmering 2. semester (Programmering 1, 2. del) beskæftiger sig med design og programmering til realisering af IT-systemer. Der fokuseres på at programmere IT-systemer af høj kvalitet i en hensigtsmæssig arkitektur med brugergrænseflade, funktionalitet og database. Arkitekturen er baseret på Spring MVC. Brugergrænsefladen udarbejdes ved hjælp af webteknologi: HTML/CSS i en klient/serverarkitektur med et databasesystem tilknyttet. Data eksponeres til brugeren ved hjælp af server-side rendering.

Der lægges vægt på godt programdesign og udvikling af systemer under hensyntagen til kvalitetssikring, herunder automatiseret unittest.

### EKSAMENSFORUDSÆTNINGER

- Deltagelse i tre gruppeprojekter på semesteret med godkendt indsats.

### LÆRINGSMÅL

- A. At kunne anvende IntelliJ til udvikling af javaprogrammer
- B. At kunne anvende Java Collection interfaces og klasser (herunder Set, Map, List, HashSet og HashMap)
- C. At kunne implementere en lagdelt arkitektur med Spring MVC
- D. At kunne anvende Spring Boot til at håndtere http GET og POST forespørgsler samt sende svar i form af dynamisk HTML indhold eller http-responskoder.
- E. At kunne anvende Thymeleaf til at rendere dynamisk html indhold.
- F. At kunne hente og indsætte data i MySQL fra Java ved brug af JDBC og JdbcTemplate
- G. At anvende sessions til at persistere data mellem http-forespørgsler
- H. At kunne implementere unit tests med JUnit og mocks efter arrange/act/assert-systematikken
- I. At kunne implementere system- og integrationstest
- J. At kunne forklare kvalitative og kvantitative egenskaber ved algoritmer og datastrukturer
- K. At kunne realisere et mindre IT-system fra krav til accepttest
- L. At kunne udføre kodereviews
- M. At kunne implementere brugergrænseflader med html og css

### PROGRESSION OG SAMMENHÆNG MED ANDRE SEMESTRE

Videreudvikling af 1. semesters basale introduktion af interface.

Minimal anvendelse af Spring Boots Dependency Injection og Inversion of Control, herunder: @Autowired, @Controller, @Service, @Repository

Afvikling af databaseforespørgsler - opsætning, design og analyse varetages i andre fagområder  
Test-design og -analyse håndteres i andre fagområder.

### LITTERATUR OG LÆREMIDLER

Skal indkøbes af de studerende:

- Ingen.

Udleveret litteratur på Fronter:

- <https://www.thymeleaf.org/documentation.html>
- <https://spring.io/guides/gs/spring-boot>

- <https://www.youtube.com/watch?v=KTBWCJPKiqk>
- <https://github.com/nicklasdean/spring-jdbc>

#### Værktøjer

- IntelliJ Ultimate Edition: <https://www.jetbrains.com/idea/download> bruges som IDE hele semestret

## BILAG: KORTLÆGNING AF STUDIEORDNINGENS LÆRINGSMÅL

### Viden

	1. Semester	2. Semester
viden om specifikation af abstrakte datatyper	A	B
viden om kriterier for program-kvalitet	(2. semester)	C, H
forståelse for abstraktionsmekanismer i moderne programmeringssprog samt forståelse for anvendelse af teori og metode inden for programmering	C, O	B

### Færdigheder

	1. Semester	2. Semester
anvende centrale metoder til at specificere og konstruere algoritmer	C, O	J
anvende centrale faciliteter i programmeringssproget til realisering af algoritmer, design-mønstre, abstrakte datatyper, datastrukturer, designmodeller og brugergrænseflader	A, C, O, P	B, C, D, E, G, M
anvende et udbredt udviklingsmiljø, herunder versionsstyringssystem og centrale softwarekomponenter/biblioteker til at designe og konstruere praksisnære applikationer baseret på en hensigtsmæssig arkitektur	D, E, F	A, C, D
anvende centrale metoder og teknikker til at realisere modeller i et databasesystem og konstruere programmer, der interagerer med en database	(2. semester)	C, F
anvende centrale metoder og teknikker til at designe og konstruere programmer der indeholder parallel afvikling og samtidighed	(2. semester)	D,
anvende tidstypiske teknikker og værktøjer til afvikling af test og kvalitetssikring samt udfærdige dokumentation i forhold til gældende standarder i professionen.	G, H, I, K	H, I

vurdere kvalitative og kvantitative egenskaber ved algoritmer og datastrukturer	(2. semester)	J
formidle praksisnære problemstillinger og løsningsmuligheder til fagfæller inden for programering	J	L

### Kompetencer

	1. Semester	2. Semester
håndtere udviklingsorienterede situationer inden for programering i mindre IT-projekter	L	A
deltage i fagligt og tværfagligt samarbejde med en professionel tilgang i relation til mindre IT-projekter med udgangspunkt i programmering	M	A
i en struktureret sammenhæng tilegne sig ny viden, færdigheder og kompetencer inden for programmeringssprog, udviklingsmiljøer, programmeringsteknikker og programdesign	(2. semester)	A, K