

Førsteårsprøve på datamatikeruddannelsen på Zealand Næstved, 2023

Virksomheden - Karsten Dahl Vandrup
Systemudvikling og Teknologi – Mikkel Nørgaard Hansen
Database og programmering – Susanne Hansen
Censor -

Indhold

Om førsteårsprøven	1
Eksamensperiode	2
Gruppearbejde og individuelle bidrag til opgaven	2
Aflevering og formalia	2
Grupper	3
Casebeskrivelse	4
Mødedatoer og deadlines	5
Immaterielle rettigheder	5
Krav til de forskellige fagområder	5
Krav til faget Virksomhed	5
Krav til faget Teknologi	6
Krav til faget Systemudvikling	6
Krav til faget Programmering	7
Om fagelementet programmering	8
Bilag 1 - Litteratur Programmering	9
Bilag 2 - Læringsmål for Programmering	10
Bilag 3 - Studieordningen om prøven	11

Om førsteårsprøven

Førsteårsprojektet er jeres første reelle eksamensopgave.

Fra nu af og op til eksamenen i juni kommer I dagligt at arbejde i jeres respektive projektgrupper. Prøveperioden skal bruges på at løse en praksisnær udviklingsopgave, skrive en god rapport og forberede jer til prøven. Jeres praktiske projekt danner grundlaget for førsteårsprøven.

Eksamensperiode

Den første del af projektet kører i perioden **8. maj - 7. juni 2023**, og perioden slutter med aflevering af jeres skriftlige rapport. Der er afsat tid til projektarbejde og vejledning i alle fag indenfor denne periode.

De efterfølgende mundtlige eksamener vil finde sted **d. 21. og 22. og 23. juni 2023**.

Det forventes, at I fortsætter arbejdet med kodning, udvikling og test mellem rapport-afleveringen den 7. juni og eksamen den 21. juni således, at I kan præsentere en forbedret version af løsningen ved eksamen.

Se [Wiseflow](#) for yderligere information og detaljer for årsprøven (være OBS på seneste afleveringstidspunkt!)

Gruppearbejde og individuelle bidrag til opgaven

Projektet skal løses i grupper på maksimum 5 personer. Hver studerende skal aktivt bidrage til både rapporten og til koden, og gruppedannelsen er bindende, når den er godkendt af underviserne. Det skal fremgå af rapporten, hvem der har ansvar for de enkelte afsnit.

Det anbefales, at hver studerende skal have hovedansvaret for mindst ét afsnit i hvert fag og skulle kunne fremvise mindst 10 meningsfulde og veldokumenterede commits i git (herunder unit tests), og mindst 7-8 sider i rapporten.

Det er den enkelte studerendes eget ansvar at bidrage bredt til projektet således, at den studerendes bidrag til rapport, kode og øvrige projekt artefakter tydeligt kan identificeres. Centrale afsnit i rapporten kan godt have flere ansvarlige.

Aflevering og formalia

Opgaven skal afleveres i Wiseflow, inkl. alle bilag og link til en åben GitHub repository, senest inden Wiseflowet lukker for aflevering.

Rapporten skal sidenummereres og der skal være en indholdsfortegnelse. Rapporten må være på maks. 40 normalsider. Dokumentation, der ikke direkte indgår som en del af rapporten, skal samles i nummererede bilag i slutningen af rapporten. Bilagene skal fremgå af indholdsfortegnelsen. Opgaven skal forsynes med en forside, hvoraf det fremgår hvilken opgave det drejer sig om, samt hvilke personer, der indgår i gruppen, og et link til projektets GitHub repository.

Se studieordningens beskrivelse af prøvens formalia i [bilag 4](#). Specifikke krav til hvert fag står opført længere nede i dette dokument.

Grupper

Gruppe 1: Sofia, Mikkel G

Gruppe 2: Oliver H, Lau

Gruppe 3: Bekir, Mikkel R, Daniel K, Marcus O

Gruppe 4: Justin, Alexander, Nicklas, Casper H

Gruppe 5: Kevin, Sercan, Emil, Jakob

Gruppe 6: Daniel N, Oliver L, Frederik

Gruppe 7: Casper W, Stefan

Gruppe 8: Lucas, Benjamin

Casebeskrivelse

Opgave: Registrering af gæster

Logistikfirmaet Intec i Haslev har behov for at registrere, hvem der får adgang til deres lager. Gennem hele dagen kommer der lastbiler, der henter og bringer varer, og som en del af deres sikkerhedsproces, skal alle chauffører registreres.

Firmaet har varehuse i både Haslev (DK) og Boston (US).

I dag skriver chaufføren sit navn og firmanavn på et papirskema, og personen der åbner for dem, tilføjer indchecknings-tidspunktet og checker deres kørekort som ID.

Firmaet ønsker en webapplikation, så chaufførerne og andre gæster kan skrive sig ind på en laptop ved indgangen. Denne indskrivning skal lagres i en database, så informationen om hvem der har været på lageret kan hentes frem, i tilfælde af at der bliver brug for det.

Lidt tekniske noter:

Transportfirmaet, som chaufførerne kommer fra, f.eks. DHL, DSV, GLS, etc., skal lagres i en separat database, og komme frem som forslag, når chaufføren registrerer sig. Dermed kan firmanavnene indtastes på forhånd.

Det skal fremgå af data i databasen, om lokationen er Haslev eller Boston. Kommunikationssproget skal derfor også matche lokationen som default, men på begge lokationer skal man også manuelt kunne vælge mellem engelsk og dansk

Intec er som alle andre virksomheder ansvarlig overfor persondata, og er underlagt bestemmelserne i GDPR.

Ekstra udfordring: Laptoppen ved indgangen til lageret har et kamera. Hvis I har tid/mod på det, så ville det være oplagt at man med dette kan affotografere kørekort, enter fra det fysiske kort, eller som skræmbillede fra kørekortsapp'en.

Krav: Applikationen skal udvikles i Java og Spring Boot.

God fornøjelse!

Mødedatoer og deadlines

I vil i forløbet få mulighed for at snakke med en repræsentant fra kunden flere gange. Det er meningen, at I skal udvikle prototyper, som kunden får et par dage inden hvert møde, og kan kommentere på til mødet (Agile Development). Der er følgende (obligatoriske) mødetidspunkter planlagt med kunden:

- Møde 1: Tirsdag 16/5, kl. 9:10
- Møde 2: Onsdag 25/5, kl 9:10 Online
- Møde 3: Fredag d. 2/6, kl 9:10

Der vil også være vejledning af alle 3 vejledere disse dage.

Immaterielle rettigheder

Zealand ejer alle rettigheder til at alle udviklede applikationer, og de skal udvikles og udgives som Open Source.

Krav til de forskellige fagområder

Krav til faget Virksomhed

I den aktuelle case indgår der persondata af forskellig art. I skal i opgaven redegøre for hvilke data der er persondata, og hvilken kategori den tilhører i forhold til GDPR.

I GDPR's paragraf 32 stilles der krav til Datasikkerhed i forbindelse med processering af persondata. Redegør for jeres overvejelser i forbindelse med overholdelsen af denne, og andre relevante GDPR paragraffer.

Note: Skal GDPR overholdes i Boston (US), eller er der andre persondatalove som der skal tages højde for?

Husk: Det er vigtigt at I argumenterer for jeres valg, så de ikke bare fremstår som postulater, men velovervejede valg.

Krav til faget Teknologi

I forhold til databaser skal den i faget anvendte teori anvendes praktisk i forhold til projektet – der skal således være et E/R diagram over programmets database, samt en dokumentation af omformningen af diagrammet til tabeldesign. Tillige skal Normalformerne indgå i dokumentationen.

I forhold til operativsystemer skal I overveje hvilke dele af operativsystemet som jeres webapplikation benytter og i hvilke kontekster. Tillige kan I overveje hvilke krav der vil være til den hardware som webapplikationen skal operere ud fra - vil der f.eks. være komponenter der stilles nogle særlige krav til.

Overvej tillige kortfattet hvordan tråde kan anvendes i jeres program – og vis et eksempel hvis I har brugt tråde i programmet.

Krav til faget Systemudvikling

I forhold til systemudvikling skal følgende krav opfyldes i forhold projektet – både i processen med udviklingsarbejdet og i forhold til dokumentationen.

- Projektplanlægningen skal være iterativ og agil – brug SCRUM metoden, og dokumenter processen ved hjælp af projektstyringsværktøjer som Trello eller lignende.
- I forhold til Objekt Orienteret Analyse skal I sørge for at undersøge og modellere problemområdet – brug værktøjer som domæne model, use-cases og system sekvens diagrammer. Sørg også for at lave brugertest og at aktivere projektstiller i udviklingsprocessen.
- I forhold til Objekt Orienteret Design skal I sørge for at designe et system der dækker problemområdet – brug packagediagram, klassediagram og sekvensdiagrammer, samt Clean Architecture til at strukturere jeres programs struktur.
- Brug design mønstre i jeres program – og gør det tydeligt hvilke I bruger.
- Overvej jeres design af brugergrænsefladen – og sørg for at teste den med projektstiller.

Husk at være tydelige omkring forskellen på analyse og design – når I skal arbejde med modellering er det vigtigt at I er fokuserede på problemområdet i analysefasen og designet af systemet i designfasen.

Det er tillige vigtigt at I reflekterer over jeres proces – så lav også et afsnit hvor I overvejer, hvad der kan gøres bedre fremover.

Krav til faget Programmering

I faget programmering skal I udvikle en it-løsning som demonstrerer, at I har lært grundlæggende objektorienteret programmering.

Krav til løsningen og rapporten

Der stilles følgende krav til løsningen:

1. I skal udvikle en konkret applikation, som lever op til de krav, som I fastsætter i samarbejde med casevirksomheden. Applikationen skal tage udgangspunkt i den kravspecifikation, der udvikles i samarbejde med casevirksomheden. Systemet skal i første omgang indeholde det, der defineres som kerne funktionaliteten i systemet.
2. Jeres implementering af applikationen skal afspejle jeres systemudviklingsarbejde, og I skal præsentere en lagdelt softwarearkitektur.
3. Applikationen skal kunne persistere data på en hensigtsmæssigt måde i forhold til den case, som I arbejder på. I rapporten skal I redegøre for jeres valg af persisteringsmekanisme.
4. Casevirksomheden vil ofte ønske funktionalitet, som ikke umiddelbart kan implementeres inden for projektets rammer. Det kunne være, at kunden ønsker noget, som vil tage for lang tid at implementere, eller som kræver eksterne ressourcer, som vi ikke har adgang til. Disse ønsker skal dokumenteres, men det er ikke et krav, at disse implementeres.
5. Vælg en eller flere testmetoder til at teste løsningen med. Beskriv hvilke testmetoder I har valgt at anvende og begrund valget. Hvert gruppemedlem skal som minimum kode nogle unit tests, og I skal skrive et afsnit om test i rapporten.
6. Beskriv de datastrukturer I har anvendt, og argumenter for jeres valg af datastrukturer.
7. Kildekoden skal afleveres både som et link til jeres GitHub repository og også som en .zip fil som bilag på WiseFlow. Repositoriet skal vise adskillige commits fra alle af gruppens medlemmer.

Om fagelementet programmering

Jeres løsning, rapporten og fremlæggelsen skal demonstrere viden, færdigheder og kompetencer som beskrevet i [studieordningens læringsmål for fagelementet programmering](#). De konkrete emner, som vi har gennemgået på første år er følgende:

1. Anvendelse af en tidssvarende udviklingsmiljø (Spring Boot, Scene Builder og IntelliJ)
2. Versionering System (Git og GitHub)
3. Unit test
4. Klasser og objekter
5. Variabler og datatyper
6. Metoder
7. Loops (for, while, foreach) og selections (if else, switch)
8. Datastrukturer (Arrays, ArrayLists, LinkedList, Set og Map)
9. Nedarvning og polimorfi
10. Graphical User Interface (GUI)
11. Algoritmer og algoritmernes effektivitet
12. Persistering af data. I tekstfil eller i database (SQLite, MySql).
13. Databaser og SQL.
14. Exception Handling
15. Web-udvikling
16. Tråde
17. Design mønstre (Singleton og Factory)

Bilag 1 - Litteratur Programmering:

1. Introduction to Java Programming, Y. Daniel Liang,

[Introduction to Java Programming and Data Structures, Comprehensive Version, 12th Edition:](#)

Liang-bogens kapitler:

2. Elementary Programming
3. Selections
4. Mathematical Functions, Characters, and Strings
5. Loops
6. Methods
7. Single-Dimensional Arrays
8. Multidimensional Arrays
9. Objects and Classes
10. Object-Oriented Thinking
11. Inheritance and Polymorphism
12. Exception Handling and Text I/O
13. Abstract Classes and Interfaces
18. Recursion
20. List, Stack, Queue
21. Set, Map
23. Sorting
32. Multithreading and Parallel Programming
34. Java Database Programming
44. Testing Using JUnit

2. Github: <https://docs.github.com/en/get-started>

3. Test: <https://www.jetbrains.com/help/idea/tests-in-ide.html>

http://webhotel4.ruc.dk/~keld/teaching/CSS_f14/Rosendahl99.pdf

4. Sortering: <https://stackabuse.com/sorting-algorithms-in-java/>

5. Stakke og køer: http://www.docjava.dk/datastrukturer/stack_queue/stack_queue.htm

6. Linked List: <https://www.javatpoint.com/java-linkedlist>

7. Dokumentation JavaDoc:

<https://www.jetbrains.com/help/idea/working-with-code-documentation.html>

8. Test: http://webhotel4.ruc.dk/~keld/teaching/CSS_f14/Rosendahl99.pdf

9. Design Patterns: <https://java-design-patterns.com/patterns/>

https://www.tutorialspoint.com/design_pattern/index.htm

10. Principper for programudvikling: <https://java-design-patterns.com/principles/>

Bilag 2 - Læringsmål for Programmering

Fagelementet beskæftiger sig med design og programmering til realisering af IT-systemer. Der fokuseres på at programmere IT-systemer af høj kvalitet i en lagdelt arkitektur med brugergrænseflade, funktionalitet og database. Løsninger konstrueres under anvendelse af værktøjer og teknikker der anvendes i professionen og der lægges vægt på at understøtte godt programdesign og udvikling af systemer af høj kvalitet.

Viden

- udviklingsbaseret viden om specifikation af abstrakte datatyper samt kriterier for programkvalitet
- forståelse for abstraktionsmekanismer i moderne programmeringssprog

Færdigheder

- anvende centrale metoder til at specificere og konstruere algoritmer samt vurdere kvalitative og kvantitative egenskaber ved algoritmer og datastrukturer
- anvende centrale faciliteter i programmeringssproget til realisering af algoritmer, designmønstre, abstrakte datatyper, datastrukturer, designmodeller og brugergrænseflader
- anvende et i professionen udbredt, integreret udviklingsværktøj, herunder versionsstyringssystem og centrale softwarekomponenter/biblioteker til at designe og konstruere praksisnære applikationer baseret på en lagdelt arkitektur
- anvende centrale metoder og teknikker til at realisere modeller i et databasesystem og konstruere programmer, der benytter en databasegrænseflade
- anvende centrale metoder og teknikker til at designe og konstruere programmer som samarbejdende processer/tråde
- anvende moderne teknikker og værktøjer til afvikling af test og kvalitetssikring samt udfærdige dokumentation i forhold til gældende standarder i professionen

Kompetencer

- håndtere udviklingsorienterede situationer inden for programmering
- deltage som en professionel programmør i udviklings- og vedligeholdelsesprojekter

· i en struktureret sammenhæng tilegne sig ny viden, færdigheder og kompetencer inden for programmeringssprog, udviklingsværktøjer, programmeringsteknikker og programdesign

Bilag 3 - Studieordningen om prøven

8.1.2. Førsteårsprøven

Førsteårsprøven udgøres af den prøve, som den studerende ifølge denne studieordning skal deltage i inden udgangen af det første studieår. Førsteårsprøven skal være bestået inden udgangen af den studerendes 1. studieår efter studiestart, for at den studerende kan fortsætte uddannelsen.

Se henvisning til den gældende bekendtgørelse om prøver i erhvervsrettede videregående uddannelser i afsnit 1.

Forudsætninger for deltagelse i prøven:

Der er et antal obligatoriske studieaktiviteter knyttet til prøven, disse er beskrevet i semesterplanen. De obligatoriske studieaktiviteter for 1. studieår skal være bestået for at gå til Førsteårsprøven.

Førsteårsprojektet, som udgør såvel bedømmelses- som eksaminations-/prøvegrundlag skal opfylde formkravene, jf. nedenfor og være afleveret rettidigt, jf. afleveringsdato på Zealands intranet.

Ikke opfyldelse af blot én eller flere obligatoriske studieaktiviteter eller korrekt aflevering af Førsteårsprojekt, som udgør den skriftlige del af prøven betyder, at den studerende ikke kan deltage i prøven, og der er brugt et prøveforsøg.

Prøvens tilrettelæggelse og indhold:

Prøven er en ekstern individuel mundtlig prøve på baggrund af et skriftligt gruppeprojekt og bedømmes efter 7-trinsskalaen.

Gruppen må højst være på 5 studerende.

Der gives én samlet karakter ud fra en helhedsvurdering af den skriftlige og den mundtlige præstation. Projektet præsenteres af projektgruppen i 10 min pr. studerende dog max. 30 min. Herefter individuel eksamination af gruppens medlemmer. Der afsættes 30 minutter pr. eksaminand inkl. votering.

Projektrapporten må maksimalt have et omfang på 40 normalsider. En normalside er 2.400 tegn inkl. mellemrum og fodnoter. Forside, indholdsfortegnelse, litteraturliste samt bilag tæller ikke med i det krævede antal sider. Bilag er uden for bedømmelse.

Ikke opfyldelse af forudsætningerne betyder, at den studerende ikke kan deltage i prøven, og der er brugt et prøveforsøg.

Formkrav:

Førsteårsprojektet skal opfylde følgende formkrav:

- Forside med angivelse af titel og rapportens omfang i antal tegn inkl. mellemrum
- Oversigt over hovedforfatter af rapportens dele
- Angivelse af sti til versionsstyringsserver, hvor kildekoden og eksekverbar kode til produktet kan hentes

Prøvens sprog er dansk eller engelsk jf. sproget for studiet, den studerende er indskrevet på.

Bedømmelseskriterier:

Bedømmelseskriterierne for prøven svarer til læringsmål for de nationale fagelementer: Programmering, Systemudvikling, Teknologi og Virksomheden. Se afsnit 2.1-2.4 i den nationale del af studieordningen.

Tidsmæssig placering:

Prøven placeres ved udgangen af 2. semester. Nærmere oplysning om tid og sted samt om aflevering af det skriftlige gruppeprojekt findes på intranet.

Prøven skal være bestået inden udgang af 1. studieår for, at den studerende kan fortsætte uddannelsen. Uddannelsen kan for den enkelte studerende dispensere fra de tidspunkter, der er fastsat for at bestå prøven, hvis det er begrundet i sygdom, barsel eller usædvanlige forhold.

ECTS-omfang:

60 ECTS