

## i Informasjon om eksamen

# EKSAMEN

**Emnekode og -navn:** ITF20319-1 19H Software Engineering og testing, skriftlig eksamen

**Dato og tid:** 26.11.19, 3 timer

**Fagansvarlig:** Mats S. Lindh

**Tillatte hjelpemidler:** Ingen

**Om eksamensoppgaven:**

Oppgavesettet består av **6** oppgaver som alle skal besvares.

**Merk 1:**

Dersom du finner at oppgaveteksten ikke gir tilstrekkelig informasjon, eller at oppgaveteksten er tvetydig, så må du gjøre dine egne forutsetninger. Disse må du i så fall presisere og begrunne.

Legg spesielt merke til hva hver enkelt oppgave ber deg besvare.

**Merk 2:**

Tidsangivelsen på oppgavene sier noe om hvor lang tid du bør sette av til hver oppgave og hvor omfattende svaret bør være. Dette er kun et utgangspunkt og ikke en fast vektning for total karakteren fra hver oppgave. Eksamens karakteren settes etter en helhetlig vurdering av den innleverte besvarelsen etter beskrivelse fra universitets- og høgskolerådet.

Dersom du får knapp tid er det bedre å ha korte beskrivelser eller skisser på punktene du ikke får tid til, fremfor å la punktene være blanke.

Det kreves ikke omfattende drøftinger i forbindelse med figurer o.l., men du bør gi kommentarer, bl.a. hvis du har gjort vesentlige valg.

Husk at begrunnelsen og beskrivelsen i svarene er viktig og derfor bør være tydelig.

Lykke til!

**Sensurfrist:** 17.12.19

Karakterene er tilgjengelige i Studentweb.

## 1 Oppgave 1 - Prosjektform (0,5t)

I denne oppgaven skal kandidaten levere utdypende definisjon av noen begreper som anvendes mye i sammenheng med smidig programvareutvikling. Bruk gjerne eksempler som kan illustrere beskrivelsen.

a)

Forklar forskjellene mellom utelukkende planbasert utvikling og utelukkende smidig utvikling.

b)

Utdyp hva disse fire begrepene beskriver og brukes til i sammenheng med smidig programvareutvikling:

- Stand-up
- Backlog
- Minimum Viable Product (MVP)
- De tre amigos

**Skriv ditt svar her...**

## 2 Oppgave 2 - Versjonskontroll (0,5t)

a)

Hva er hensikten med branches i et versjonskontrollsystem?

b)

Hvilke fordeler gir versjonskontroll oss med hensyn til Containerization og Orchestration?

**Skriv ditt svar her...**

## 3 Oppgave 3 - Avhengigheter (0,4t)

a)

Forklar hvorfor det er viktig å håndtere avhengigheter ved hjelp av et avhengighets- eller pakkesystem i programvareprosjekter.

b)

Hva er oppgaven til en låsefil ("lock file") som benyttes i mange avhengighets- og pakkesystemer, og hvilke fordeler gir det?

**Skriv ditt svar her...**

#### 4 Oppgave 4 - Testing (0,4t)

a)

Beskriv kort forskjellene og forholdet mellom enhetstester, integrasjonstester og systemtester.

b)

Forklar hva Red Green Refactor-metoden går ut på og hva målet med den er.

**Skriv ditt svar her...**

#### 5 Oppgave 5 - Praktisk testing (0,6t)

Et skuddår forekommer vanligvis ved hvert årstall som er delelig på fire, med mindre årstallet samtidig er delelig på 100 - og ikke er delelig på 400.

1996 var et skuddår (delelig på fire), 1900 var ikke et skuddår (delelig på fire, men også delelig på 100), men 2000 var imidlertid et skuddår (delelig på fire, delelig på hundre, men også delelig på 400).

Anta at du har en funksjon eller metode som gir **true** eller **false** tilbake, avhengig av om det er skuddår i et gitt år. Metoden er definert som:

**bool isLeapYear(int year)**

Lag et sett med tester (beskrevet som pseudokode eller i valgfritt programmeringsspråk) som viser at funksjonen oppfører seg som beskrevet. Du kan anta at funksjonen eller metoden allerede er implementert og er korrekt i henhold til spesifikasjonen over. Det er kun testmetodene som er relevante. Import av testrammeverk etc. trenger du ikke tenke på.

Du har direkte tilgang til metodene **assertTrue** og **assertFalse**.

**Skriv ditt svar her...**

#### 6 Oppgave 6 - Arkitektur (0,6t)

Beskriv hvert av de fem prinsippene som ligger bak akronymet SOLID.

**Skriv ditt svar her...**