Introduction

UNIVERSITETET I OSLO

Det matematisk-naturvitenskapelige fakultet

Skriftlig eksamen, IN3240/IN4240 - Testing av programvare
2026 VÅR
Varighet: 6. juni kl. 15:00 - 19:00
Tillatte hjelpemidler: Ingen.

Det er viktig at du leser denne forsiden før du starter. Hvis du trenger å bruke zoomfunksjonen, kan du holde nede ctrl og klikke + eller - på det numeriske tastaturet.

Oppgaven består av fire deler.

Du kan skrive på både engelsk og norsk.

Del 1 - Flervalgsoppgaver/matchingsoppgaver

- 6 flervalgsoppgaver
- 7 matching-oppgaver
- Maksimumspoengsum for del 1: 25 poeng.
- Oppgaver og svaralternativer kommer i random rekkefølge.

Del 2 - Spesifikasjonsbaserte teknikker

- 3 caser:
- EP/BVA: 10 poeng.
- Beslutningstabeller: 10 poeng.
- Tilstand-overgangs testing: 10 poeng.
- Maksimumspoengsum for del 2: 30 poeng.

Del 3 - Tekstoppgaver

- 5 tekstoppgaver som skal besvares kort og nøyaktig.
- Maksimumspoengsum for del 3: 25 poeng.

Del 4 - Essay

- Maksimumspoengsum for del 4: 20 poeng.

Hvis noen oppgaver er uklare, kan du gjøre dine egne antagelser. Gjør i så fall rede for dem.

1 Matching

Match elementene i listen over risikoer med enten produkt- eller prosjekt-risiko:

	Produktrisiko	Prosjektrisiko
En defekt kan tillate hackere å få administrative rettigheter.	○	○
Programmet passerer ikke valideringen.	○	○
Feil beregning av gebyrer (i programmet under test).	○	○
En leverandør kan ikke levere en systemkomponent i tide.	○	○
En feilprioriteringsprosess kan overbelaste utviklingsteamet.	○	○
Et kompetansegap kan oppstå ved bruk av et nytt automatiseringsrammeverk.	○	○

Maks poeng: 3

2 Matching

Organiser listen av testaktiviteter i riktig rekkefølge:

	1	2	3	4	5
Testavslutningsaktiviteter	○	○	○	○	○
Analyse og design	○	○	○	○	○
Evaluering av exit-kriterier og rapportering	○	○	○	○	○
Planlegging	○	○	○	○	○
Implementasjon og utførelse	○	○	○	○	○

Maks poeng: 2.5

3 Multiple choice

Fullfør setningen slik at den gir den mest presise definisjonen av alpha-testing.

Alpha-testing ...

Velg ett alternativ:

- ○ ... er nyttig for å teste programvare utviklet for en spesifikk kunde eller bruker.
- ○ ... utføres av sluttbrukere hos utviklerne.
- ○ ... utføres av sluttbrukere på deres eget sted.
- ○ ... utføres av et uavhengig testteam.

Maks poeng: 1

4 Matching

ISTQB skiller mellom begrepene testtype og testnivå.
Avgjør hva som er en testtype og hva som betegner et testnivå.

	Testtype	Testnivå
Integrasjonstesting	○	○
Akseptansetesting	○	○
Systemtesting	○	○
Endringsrelatert testing	○	○
Ikke-funksjonell testing	○	○
Enhetstesting	○	○
Funksjonell testing	○	○
Strukturbasert testing	○	○

Maks poeng: 4

5 Matching

De spesifikasjonsbaserte (svart-boks) testteknikkene kan brukes på flere av områdene nedenfor. Du skal imidlertid matche dem slik at alle teknikkene blir brukt på det mest egnede området.

	Tilstand/overgangs-testing	Testing basert på brukstilfeller	EP og BVA*	Beslutningstabell-testing
Testing av brukerhistorier.	○	○	○	○
Testing av et frankeringssystem for brevpost	○	○	○	○
Testing av funksjonaliteten til innloggingsautomat på en flyplass	○	○	○	○
Testing av et program som, ut fra ulike opptakskriterier, behandler søknader til et universitetsstudium.	○	○	○	○

** EP og BVA: Ekvivalenspartisjonering (ekvivalensklasseinndeling) og grenseverdianalyse*

Maks poeng: 2

6 Matching

Avgjør hvilke testaktiviteter som primært utføres av testeren og hvilke som primært utføres av testlederen.

	Testleder	Tester
Evaluere testresultatene mot exit-kriteriene	○	○
Koordinere teststrategi og plan med prosjektledere	○	○
Analysere, gjennomgå og vurdere brukerkrav	○	○
Anskaffe og utarbeide testdata	○	○
Designe testtilfeller	○	○
Bestemme hva som skal automatiseres og i hvilken grad	○	○
Samle inn testresultater og rapportere om testprogresjon	○	○

Maks poeng: 3.5

7 Matching

Match følgende prosesser med tilhørende aktiviteter:

	Gruppere tester i skript	Formidle testresultatene til interessentene	Fastsette omfanget og målene for testingen	Omgjøre testmålene til testbetingelser
Test rapportering	○	○	○	○
Test analyse	○	○	○	○
Test implementasjon	○	○	○	○
Test planlegging	○	○	○	○

Maks poeng: 2

8 Multiple choice

Hvilken av disse påstandene om funksjonell testing er sann?

Velg ett alternativ

- Funksjonell testing er kraftigere enn statisk testing siden du faktisk kjører systemet og ser hva som skjer.
- Strukturell testing er viktigere enn funksjonell testing da den tar for seg koden.
- Inspeksjon er en form for funksjonell testing.
- Funksjonell testing er nyttig gjennom hele livssyklusen og kan brukes av bedriftsanalytikere, testere, utviklere og sluttbrukere.

Maks poeng: 1

9 Multiple choice

Hva er generelt de potensielle fordelene ved å bruke verktøy for å støtte testing?

Velg ett alternativ

- Større repeterbarhet av tester, reduksjon i repeterende arbeid, objektiv vurdering.
- Større kvalitet på kode, reduksjon i papirarbeid, færre innvendinger mot testene.
- Større responsivitet for brukere, reduksjon av antall kjørte tester.
- Større kvalitet på kode, reduksjon i antall testere som trengs, bedre mål for testing.

Maks poeng: 1

10 Multiple choice

ISTQB etablerer en grunnleggende testprosess der testplanlegging skjer tidlig i prosjektet, mens testutførelse skjer senere.

Hvilke av de følgende elementene i testplanen, angitt under testplanlegging, blir vurdert under testutførelsene?

Velg ett alternativ

- Exit-kriterier
- Testoppgaver
- Trening av testteam
- Testomgivelsesbehov

Maks poeng: 1

11 Multiple choice

Hvilken påstand om statisk analyse er sann?

Velg ett alternativ

- Statisk analyse finner alle feil.
- Ved hjelp av statisk analyse kan man finne mangler som er vanskelige å finne med dynamisk testing.
- Brukt riktig gjør statisk analyse funksjonstesting overflødig.
- Kompilering er ikke en form for statisk analyse.

Maks poeng: 1

12 Multiple choice

Ikke-funksjonell testing inkluderer ...

Velg ett alternativ

- ... testing for å se hvor systemet ikke fungerer riktig.
- ... testing av kvalitetsegenskapene til systemet, inkludert pålitelighet og brukervennlighet.
- ... å teste en systemfunksjon ved å bruke bare den programvaren som er påkrevd for den funksjonen.
- ... testing for å få brukergodkjenning for systemet.

Maks poeng: 1

13 Matching

Hvilke av følgende kan være et mål på dekningsgrad for **tilstand-overgangstesting (state transition testing)?**

	Ikke mål for dekningsgrad	Mål for dekningsgrad
Responstiden for hver transaksjon.	○	○
Antall grenser som er testet.	○	○
Antall tilstander (states) som er nådd.	○	○
Antall overganger (transitions) som er passert.	○	○

Maks poeng: 2

14 Decision table

I denne oppgaven skal du lage tabeller. Bruk tabell-verktøyet i svareditoren til å lage tabellene.



Lånesøknad

For å bremse boligprisene og begrense innbyggernes lånebelastning, har stortinget bestemt seg for å stramme inn bankenes *utlånsforskrift.* Kravet til egenkapital ved kjøp av bolig skal igjen økes til 15%. I tillegg stilles det krav til stabil inntekt, samt en låneramme på maskimalt 5 ganger årsinntekten.

En bank som tilbyr boliglån har et dataprogram til å vurdere lånesøknader automatisk. For å imøtekomme den nye utlånsforskriften, må følgende krav til jobb, inntekt og egenkapital innfris for at lånesøknaden skal innvilges:

- Fast jobb.
- Minimum 15% egenkapital (15% av eiendomsverdi).
- Maksimalt lånebeløp er 5 ganger årsinntekten.

Bankene har imidlertid en fleksibilitetskvote som gir dem anledning til å avvike fra ett eller flere av kravene i utlånsforskriften for 10% av lånesøknadene.
For lånesøkere uten fast jobb, men som tilfredsstiller de øvrige betingelsene, er det derfor fortsatt mulighet til å få lån. Det krever imidlertid at lånesøknaden behandles manuelt. I dette tilfelle skal programmet tildele lånesøkeren en saksbehandler. Hvis derimot kravet til egenkapital ikke er tilfredsstilt, til tross for at lånesøkeren har fast inntekt, og lånebeløpet er innenfor lånerammen, skal programmet sende lånesøkeren informasjon om bankens kapitalsparingsprogram. For lånesøknader med søknadsbeløp som er høyere enn 5 ganger årsinntekten, skal banken gi et redusert lånetilbud som er innenfor rammene på 5 ganger årsinntekten, forutsatt at de andre to betingelsene er innfridd.

a) Tegn en beslutningstabell som viser alle mulige kombinasjoner av sannhetsverdier som betingelsene kan ha. Beslutningstabellen skal inneholde en handlingsdel.

b) Reduser antall regler så mye som mulig uten at noe informasjon går tapt. Forklar hvordan og hvorfor du foretok reduksjonen.

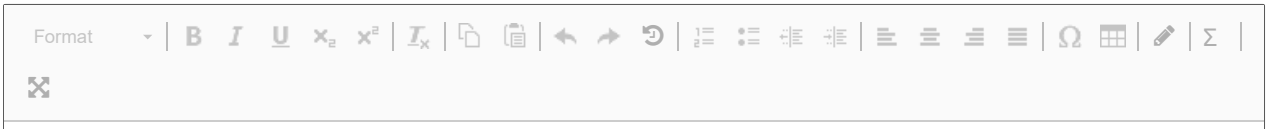
c) Gitt tre testtilfeller:
1. En lånesøknad der alle tre betingelser er tilfredsstilt.
2. En lånesøknad der søkeren er uten fast jobb, men tilfredsstiller de to andre betingelsene.
3. En lånesøknad med et søknadsbeløp som er høyere enn 5 ganger årsinntekten, men der søkeren har fast jobb og tilstrekkelig egenkapital.

Trenger vi flere testtilfeller? Hvis ja, hvilke?

Det ser imidlertid ut til at mange unge sliter med å komme inn på boligmarkedet. Stortinget har derfor bestemt av kravet til egenkapital skal reduseres til 10% av eiendommens markedsverdi for lånesøkere under 30 år.

c) Hvor mange forskjellige kombinasjoner av sannhetsverdier får vi hvis vi legger til betingelsen "under 30 år"? Hvordan øker antallet sannhetsverdikombinasjoner i forhold til antall betingelser? Hvordan påvirker dette bruken av beslutningstabeller?

Skriv ditt svar her...



Maks poeng: 10

15 Equivalence Partitioning / Boundary Value Analysis

Trinnskatt

I Norge betaler skattebetalere flere typer skatt. En av disse er såkalt *trinnskatt*, der skatteprosenten er progressiv i forhold til inntekten. Trinnskatten beregnes på følgende måte:

Trinn 0: Av inntekten under 217 400 kr betaler du ikke trinnskatt.
Trinn 1: Av Inntekten mellom 217 400 og 306 049 kr betaler du 1,7% trinnskatt.
Trinn 2: Av inntekten mellom 306 050 og 697 149 kr betaler du 4% trinnskatt.
Trinn 3: Av inntekten mellom 697 150 og 937 900 kr betaler du 13,6% trinnskatt.
Trinn 4: Av inntekten mellom 937 901 og 1 350 000 kr betaler du 16,6% trinnskatt.
Trinn 5: Av inntekten fra og med 1 350 001 kr betaler du 17,6 % trinnskatt.

La oss si at du tjener 400 000 kr i 2025. Det betyr at du betaler 1,7 % skatt på inntekten din mellom 217 400 og 306 049 kr, og 4 % skatt på inntekten din mellom 306 050 og 400 000 kr.

Tenk deg at du skal teste en metode (funksjon) i et program som blant annet beregner trinnskatten. På grunnlag av skattebetalerens inntekt, angitt som heltall, skal metoden berenge trinnskatten vedkommende skal betale.

a) Del mengden av input-verdier inn i ekvivalensklasser (equivalence partitions), både gyldige og ugyldige.

b) Hvilke to betingelser må delmengdene i en ekvivalensklasseinndeling (equivalence partitioning) oppfylle for at inndelingen skal være gyldig? Vis at inndelingen du gjorde under punkt a) oppfyller betingelsene.

c) Hvilken *annen* betingelse må *programmet* oppfylle for å kunne bruke ekvivalensklasseinndeling som testdesignteknikk?

d) Velg input-verdier til testene slik at du oppnår 100% ekvivalensklassedekning (equivalence partition coverage) med færrest mulig tester (en input-verdi pr. test).

Ekvivalensklasseinndeling brukes gjerne sammen med grenseverdianalyse (boundary value analysis).

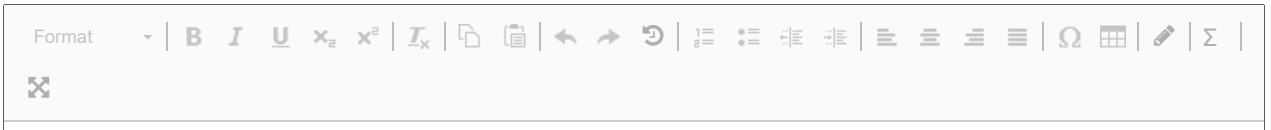
e) Hvilken betingelse må mengden av input oppfylle for at vi skal kunne bruke grenseverdianalyse?

f) Bestem eventuelle grenseverdier som bør testes.

g) Hvorfor tester vi grenseverdier?

h) Hvorfor tester man gjerne *både* grenseverdier *og* andre verdier fra eksvivalensklassene, når det strengt tatt ikke øker ekvivalensklassedekningen?

Skriv ditt svar her



Words: 0

Maks poeng: 10

16 State transition testing

Tilstand-overgangstesting er basert på det som i matematikken kalles for en endelig tilstandsmaskin, der oppførselen til et program er beskrevet som et tilstandsdiagram.

En restaurant i Oslo har utviklet en QR-meny-app. Før de tar den i bruk, ønsker de å teste funksjonaliten. I den vedlagte pdf-filen ser du et tilstandsdiagram som beskriver programmet.
NB! Du skal ikke vurdere om diagrammet inneholder feil, men bruke det nøyaktig som det er.

a) Beskriv et testtilfelle som en sti fra start (tilstand 1) til slutt (tilstand 12) som er innom flest mulig tilstander. (Du kan skrive tilstandene etter hverandre, adskilt med komma eller bindestrek.) Hvis det er flere stier som er innom det høyeste antallet tilstander, skal du velge den korteste stien.
NB! Du skal kun følge gyldige overganger fra en tilstand til den neste.

- Hva er tilstandsdekningen (*state coverage*) til dette testtilfellet?
- Hva er overgangsdekning *(transition coverage*) til dette testtilfellet?

b) Er det mulig å dekke alle gyldige overganger i ett enkelt testtilfelle basert på tilstandsdiagrammet? Hvis ikke, hvor mange testtilfeller trenger du for å dekke alle overganger? Begrunn svarene og angi stien(e) som trengs for å dekke alle.

c) Hva er den høyeste overgangsdekningen (*transition coverage*) du kan oppnå i ett enkelt testtilfelle? Angi stien og beregn overgangsdekningen.

d) Hvilken dekningstype, tilstandsdekning eller overgangsdekning, er sterkest? Finnes det en sammenheng mellom dem, og i så fall, hvilken sammenheng?

Skriv ditt svar her

Format

Words: 0

Maks poeng: 10

17 Text assignment

Ett av de syv testprinsippene kalles

Feilslutning vedrørende fravær av feil (Absence-of-errors fallacy).

Hva forteller dette prinsippet oss, og hvordan bør det påvirke testaktivitetene?

Skriv ditt svar her

Format

Words: 0

Maks poeng: 4

18 Text assignment

Hvilke andre testprinsipper har vi, ifølge ISTQB?

Skriv ditt svar her

Format

Words: 0

Maks poeng: 6

19 Text assignment

a) Forklar hva et statisk analyseverktøy er og hensikten med verktøyet.

b) Statisk analyseverktøy brukes hovedsakelig av én gruppe personer. Hvilken?

c) Gi minst tre eksempler på typer informasjon som statiske analyseverktøy kan gi.

Skriv ditt svar her...

Format

Words: 0

Maks poeng: 5

20 Text assignment

Uavhengig testing, hvor testing utføres av personer som ikke har vært involvert i utviklingen av det som skal testes, har sine fordeler, men også sine ulemper.

a) Nevn minst to fordeler med uavhengig testing.

b) Nevn minst én ulempe ved uavhengig testing.

c) Gi eksempler på minst fire ulike grader av testuavhengighet.

d) Ranger de fire testnivåene i forhold til hvor utbredt uavhengig testing er, fra minst utbredt til mest utbredt.

Skriv ditt svar her

Format

Words: 0

Maks poeng: 5

21 Text assignment

a) Forklar hva *utforskende testing* (exploratory testing) er.

b) Hva kan begrense nytteeffekten av utforskende testing?

c) Nevn minst to situasjoner der det kan være hensiktsmessig å velge utforskende testing som teststrategi.

Skriv ditt svar her

Format

Words: 0

Maks poeng: 5

22 # Essay

Statisk testing

Statisk testing spiller en viktig rolle innen programvareutvikling. Vi deler den inn i to hovedtyper, *Reviews* (granskninger) og *Statisk analyse*.

Skriv en sammenhengende tekst om statisk testing, der du besvarer så mange som mulig av spørsmål nedenfor. Du står imidlertid fritt til å skrive mer om emnet enn de momentene som er fremhevet i spørsmålene.

- Hva er *statisk testing*?
- Hva er formålet og fordelene med statisk testing?
- Hvor i utviklingsprosessen utfører vi statisk testing?
- Hvilke(t) testprinsipp(er) oppfordrer indirekte til gjennomføring av statisk testing?
- Hva skiller statisk testing fra dynamisk testing?
- Kan statisk testing erstatte dynamisk testing?
- Hvilke review-typer har vi, og hva kjennetegner de ulike typene? Bruk gjerne eksempler.
- Hvilke roller har deltagerne i de ulike reviews-typene, og hva er rollenes ansvarsområder?
- Hvilke fem faser inngår i et formelt review?
- ...

Skriv ditt svar her

Format

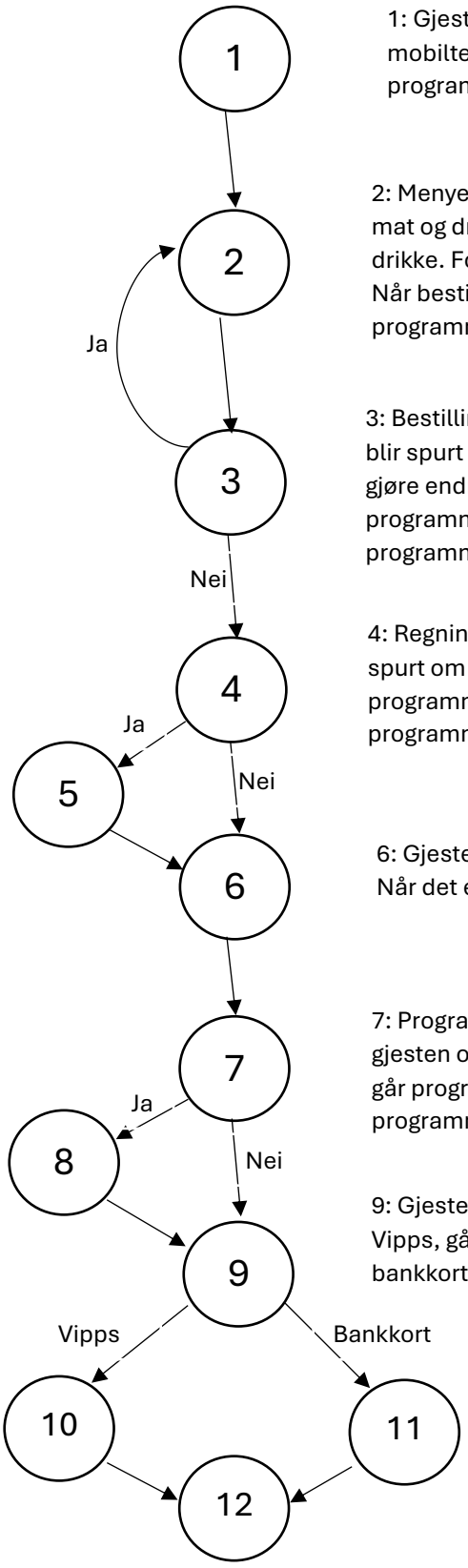
Words: 0

Maks poeng: 20

Question 16
Attached



QR-meny



1: Gjesten skanner QR-koden med mobiltelefonen. Hvis vellykket skanning, går programmet til tilstand 2.

2: Menyen åpnes i browseren. Gjesten velger mat og drikke ved å klikke på valgte retter og drikke. For hvert valg, velges ønsket antall. Når bestillingen bekreftes ferdig, går programmet til tilstand 3.

3: Bestillingen vises på skjermen, og gjesten blir spurt om hen ønsker å bestille mer, eller gjøre endringer i bestillingen. Hvis ja, går programmet tilbake til tilstand 2. Hvis nei, går programmet til tilstand 4.

4: Regningen vises på skjermen, og gjesten blir spurt om hen vil legge til dricks. Hvis ja, går programmet til tilstand 5. Hvis nei, går programmet til tilstand 6.

5: Gjesten legger til dricks. Når det er gjort, går programmet til tilstand 6.

6: Gjesten blir bedt om å skrive inn bordnummer. Når det er gjort, går programmet til tilstand 7.

7: Programmet viser regningen på skjermen og spør gjesten om hen ønsker kvittering på epost. Hvis ja, går programmet til tilstand 8. Hvis nei går programmet til tilstand 9.

8: Gjesten skriver inn epost-adresse. Når det er gjort, går programmet til tilstand 9.

9: Gjesten skal velge betalingsform. Hvis gjesten velger Vipps, går programmet til tilstand 10. Hvis gjesten velger bankkort, går programmet til tilstand 11.

10: Gjesten betaler med Vipps. Når betalingen er utført, går programmet til tilstand 12.

11: Gjesten betaler med kort. Når betalingen er utført, går programmet til tilstand 12.

12: Programmet viser følgende på skjermen:

a) Kvittering med bekreftelse på at betaling en er utført.
b) Estimert ventetid
c) Ønsker gjesten et godt måltid.