

Terminprøve

Informasjonsteknologi 2

1. desember 2017

Informasjon om prøven

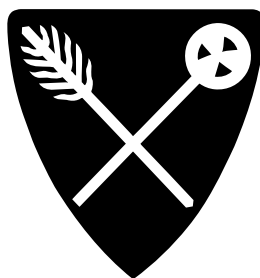
Prøven varer i 5 timer. Alle hjelpemidler er tillatt, bortsett fra internett og andre verktøy som kan brukes til kommunikasjon. Tillatte nettsider er w3schools.com samt itslearning for elektronisk innlevering.

Det hører vedlegg til oppgave 3. Disse finner du i mappen *Fugleinformasjon*.

Du skal svare på alle oppgavene, og alle deloppgaver i de tre. Unntaket er oppgave 3d, som er en valgfri utvidelse av oppgaven dersom man har tid. Der oppgaveteksten ikke sier noe annet, kan du fritt velge fremgangsmåte. Dersom oppgaven krever én bestemt løsningsmetode, vil også en alternativ metode kunne gi noe uttelling. Fremgangsmåte og forklaring vil belønnes også der resultatet ikke er riktig (koden ikke kjører/koden gjør noe annet enn spesifisert).

Oppgavene skal leveres som én zipfil som inneholder navnet ditt, for eksempel *terminproveAsta.zip*.

Karakteren blir fastsatt etter en samlet vurdering. Se neste side for vurderingskriterier.



Schola Osloensis
siden 1153

Vurderingskriterier

Oppgave 1 og 2 gir sammen like mye uttelling som oppgave 3 alene. Karakteren blir satt etter en samlet vurdering der det, i tillegg til uttelling for deloppgavene, blir vurdert i hvilken grad du

- viser kodeforståelse og velger smarte løsninger
- har en oversiktlig kode
- ser sammenhenger i faget, er oppfinnsom og kan ta i bruk fagkunnskap i nye situasjoner
- forklarer fremgangsmåte og begrunner løsningene dine dersom det spørres etter dette
- kan planlegge IT-løsninger med relevante teknikker og verktøy og lage relevant dokumentasjon i koden
- spesifiserer og tar hensyn til krav til brukergrensesnitt for et program

Kjennetegn på måloppnåelse

Relevante deler av pensum er markert i gult.

Kjennetegn på måloppnåelse ved sluttvurdering i REA3015 Informasjonsteknologi 2 – gjelder for sentralt gitt skriftlig eksamen

	Karakteren 2	Karakterene 3 og 4	Karakterene 5 og 6
Problemløsning	Eleven – programmerer løsninger med enkle variabler, og bruker enkle valgsetninger og tar i bruk standardfunksjoner og/eller metoder. – planlegger en enkel applikasjon og lager dokumentasjon for en slik	Eleven – programmerer løsninger med enkle variabler, valg og løkker og lager og bruker egne funksjoner – planlegger IT-løsninger med relevante teknikker og verktøy, lager relevant dokumentasjon og begrunner valg av programmeringsløsninger og valg av datatyper	Eleven – programmerer løsninger med indekserte variabler, lagrer og henter fram variabelverdier, bruker og velger løkker, egne funksjoner og setter sammen delprogram – spesifiserer krav til IT-løsninger, gjør rede for hvordan disse utvikles, og lager relevant dokumentasjon og vurderer ulike programmeringsløsninger
Brukergrensesnitt	Eleven – bruker og tilpasser digitale objekter av type tekst, tall, bilder, grafikk, animasjon, film og lyd – lager enkle multimediale brukergrensesnitt uten bruk av programmert kode – tar hensyn til spesifiserte krav til løsning	Eleven – lager, tilpasser og bruker digitale objekter av type tekst, tall, bilder, grafikk, animasjon, film og lyd og begrunner valg av filformater – planlegger og utvikler hendelsesstyrte multimediaapplikasjoner – spesifiserer og tar hensyn til krav til brukergrensesnitt for en multimediaapplikasjon	Eleven – utvikler og bruker relevante digitale objekter av type tekst, tall, bilder, grafikk, animasjon, film og lyd og vurderer og bruker ulike relevante objekter og filformater. – bruker programmeringsspråk i utvikling av multimediaapplikasjoner – vurderer multimediaapplikasjoner med hensyn til brukergrensesnitt og funksjonalitet
Verktøy og begreper	Eleven -leser og forstår dokumentasjon og kode -velger relevante utviklings- og planleggingsverktøy	Eleven -bruker dokumentasjon og kode -gjør rede for hensikten med objektorientert programmering -bruker relevante teknikker i utviklings- og planleggingsverktøy	Eleven -bruker relevant dokumentasjon og kode -vurderer nytten av objektorientert programmering og begrepene klasse, objekt og arv -bruker relevante teknikker i utviklings- og planleggingsverktøy og kjenner verktøyenes muligheter.

Karakteren 1 uttrykker at eleven viser lavere måloppnåelse enn det som er gjort greie for ovenfor.

Oppgave 1

Alle svar i denne oppgaven kan enten skrives til nettsiden eller til konsollen.

a) Skriv kode som genererer et tilfeldig heltall i området $[-10, 10]$ og returnerer om tallet er positivt, negativt eller lik null.

Eksempel: dersom det tilfeldige tallet er lik -3 returneres *Tallet er -3. Tallet er negativt*

b) Skriv kode der du løkker gjennom tallene fra 1 til 100 og returnerer summen av tallene.

c) Lag en funksjon som tar inn et navn og returnerer antall tegn i navnet og den første bokstaven i navnet. Kjør funksjonen for ett navn.

Eksempel: `navnFunksjon('Asta')` returnerer *Navnet har 4 tegn og den første bokstaven er A.*

d) Lag en funksjon som tar inn en lengde i miles returnerer lengden i km.

Opplysning: 1 mile \approx 1.61 km.

Eksempel: `omgjoringFunksjon(2)` returnerer *2 miles = 3.22 km*

Oppgave 2

Du velger selv om du vil lage én eller flere nettsider for denne oppgaven. Dersom du velger å presentere alle oppgaver på samme side skal det komme tydelig frem hvilken oppgave du svarer på, f.eks. ved bruk av `<div>` og kantlinjer. Alle svar i denne oppgaven skal skrives til nettsiden.

a) Lag et tekstelement som bytter bakgrunnsfarge og skriftstørrelse når musepekeren beveger seg over det og som bytter tilbake til opprinnelig når musepekeren ikke lenger er over det.

b) Lag et tekstelement der tekstinnholdet er musepekerens posisjon på nettsiden.

c) Lag en applikasjon der brukeren kan skrive inn en avstand i m og få den omgjort til km, miles eller yards (1 yard \approx 0.91 m). Applikasjonen skal bestå av et number-felt til å skrive inn avstanden, en måte å velge mellom km, miles eller yards på, en knapp for å kjøre beregningen og et felt der resultatet skrives ut. Det skal ikke være mulig å skrive inn en negativ avstand. Gi svaret med to desimaler.

d) Lag en applikasjon der brukeren velger to tall, X og Y, mellom 1 og 15 i et skjema og får skrevet ut gangetabellen til og med $X \cdot Y$.

Eksempel: Dersom brukeren velger tallene 3 og 12, skrives det ut en multiplikasjonstabell med kolonner fra 1 til 3 og rader fra 1 til 12.

Oppgave 3

Studenter ved NTNU tar ofte faget Fugletitting dersom de er på feil side av Lånekassa eller ønsker et lett semester med mye reising og lite studier. Eksamen i faget er digital og går ut på at man får opp et bilde av en fugl sammen med tre alternativer for hvilken fugl det er. Studenten får 1 poeng for korrekt svar, -1 poeng for feil svar og 0 poeng for å ikke svare.

For å gjøre det bra på eksamen burde man ha sett mange av fuglene fra før. Oppgaven går derfor ut på å lage en treningsside for studenter. Oppgaven er delt i deloppgaver som er presentert på neste side.

Krav til funksjonalitet:

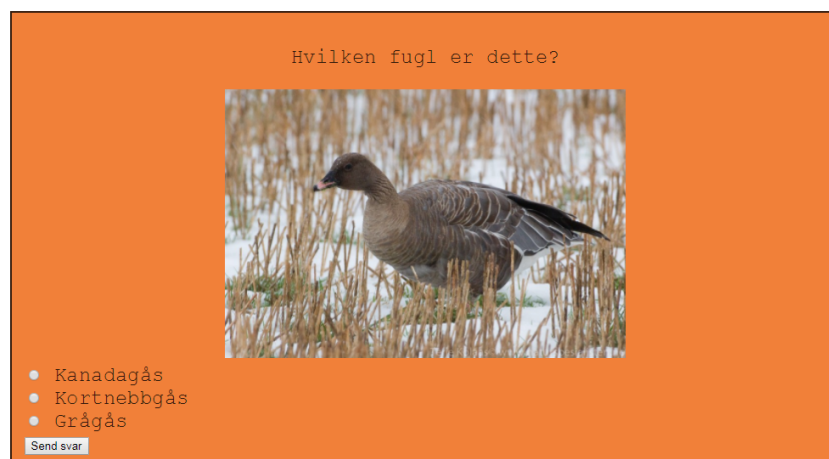
For full måloppnåelse skal studenten få presentert én fugl av gangen, med tre svaralternativer. Svaralternativene skal legges inn som radioknapper. Når ett spørsmål er besvart, skal et nytt spørsmål komme opp.

Dersom du ikke får til å presenitere én og én fugl, kan du legge ut alle de fire fuglene og svaralternativene til disse sammen, med en Sjekk svar-knapp nederst. Dette vil også gi uttelling.

Når studenten har svart på alle spørsmålene, skal poengsummen vises.

Krav til stilsetting:

Området på nettsiden der informasjonen, spørsmålene og resultatet presenteres skal være mindre enn selve nettleservinduet og skal avgrenses med en kantlinje og ha en egen bakgrunnsfarge. Se figurene under for skisse av en mulig løsning:



a) Skriv pseudokode for hvordan du tenker å løse oppgaven. Denne kan skrives som kommentar i koden eller som en egen .txt-fil.

b) Lag en startside der informasjon om eksamen vises (se fila eksamensinfo.txt) og der studenten skriver inn navnet sitt. Når studenten klikker på Start treningseksamen skal du lagre navnet på eksamenskandidaten i en variabel (denne får du bruk for senere, så pass på at den eksisterer utenfor funksjonen også) og erstatte innholdet på nettsiden med første spørsmål.

c) Fullfør hele treningssiden ved å legge inn spørsmål om fire fugler. Du velger selv hvilke fulger og svar-alternativer, og det ligger en liste over fuglenavn i filen fugler.txt. Når kandidaten har svart på alle spørsmålene skal du skrive ut teksten

Hei, navn. Du fikk poeng riktige.

der *navn* erstattes med navnet som ble skrevet inn på starten av oppgaven og *poeng* erstattes med antall poeng oppnådd.

d) **Eventuell** ekstraoppgave (dersom du har tid til overs, denne vil kun telle positivt!)

Denne koden skal du ikke skrive inn i din opprinnelige oppgave 3. Hvis du velger å besvare oppgaven, lagre en kopi av oppgave 3 der du implementerer utvidelsen. Dette for å sikre at du har svart på oppgave a) - c).

I den reelle eksamen i Fugletitting får studentene kun 30 sekunder på å gjette fuglen. Dersom tida går ut, registreres det som blankt svar. Implementer dette i koden du har laget.

Husk å levere heldagsprøven som én zip-fil på itslearning!