



KANDIDAT

**336**

PRØVE

# MAS234 G Innebygde datasystemer for mekatronikk

---

Emnekode	MAS234
Vurderingsform	Skriftlig eksamen
Starttid	18.12.2020 08:00
Sluttid	18.12.2020 12:00
Sensurfrist	07.01.2021 22:59
PDF opprettet	04.11.2021 14:39

---

Oppgave	Tittel	Oppgavetype
i	MAS234 H-20 ord forside	Dokument
1	MAS234_2020h - Oppgave 1	Flervalg
2	MAS234_2020h - Oppgave 2	Flervalg
3	MAS234_2020h - Oppgave 3	Fyll inn tekst
4	MAS234_2020h - Oppgave 4	Flervalg
5	MAS234_2020h - Oppgave 5	Paring
6	MAS234_2020h - Oppgave 6	Fyll inn tekst i bilde
7	MAS234_2020h - Oppgave 7	Nedtrekk
8	MAS234_2020h - Oppgave 8	Flervalg
9	MAS234_2020h - Oppgave 9	Fyll inn tall
10	MAS234_2020h - Oppgave 10	Flervalg
11	MAS234_2020h - Oppgave 11	Flervalg
12	MAS234_2020h - Oppgave 12	Fyll inn matematikk
13	MAS234_2020h - Oppgave 13	Flervalg
14	MAS234_2020h - Oppgave 14	Fyll inn tekst
15	MAS234_2020h - Oppgave 15	Flervalg
16	MAS234_2020h - Oppgave 16	Flervalg
17	MAS234_2020h - Oppgave 17	Flervalg
18	MAS234_2020h - Oppgave 18	Flervalg
19	MAS234_2020h - Oppgave 19	Flervalg

20	MAS234_2020h - Oppgave 20	Flervalg
21	MAS234_2020h - Oppgave 21	Fyll inn tekst
22	MAS234_2020h - Oppgave 22	Flervalg
23	MAS234_2020h - Oppgave 23	Sant/usant
24	MAS234_2020h - Oppgave 24	Sant/usant
25	MAS234_2020h - Oppgave 25	Flervalg
26	MAS234_2020h - Oppgave 26	Flervalg
27	MAS234_2020h - Oppgave 27	Flervalg
28	MAS234_2020h - Oppgave 28	Flervalg
29	MAS234_2020h - Oppgave 29	Sant/usant
30	MAS234_2020h - Oppgave 30	Flervalg
31	MAS234_2020h - Oppgave 31	Flervalg
32	MAS234_2020h - Oppgave 32	Flervalg
33	MAS234_2020h - Oppgave 33	Flervalg
34	MAS234_2020h - Oppgave 34	Flervalg
35	MAS234_2020h - Oppgave 35	Paring
36	MAS234_2020h - Oppgave 36	Flervalg
37	MAS234_2020h - Oppgave 37	Flervalg
38	MAS234_2020h - Oppgave 38	Flervalg
39	MAS234_2020h - Oppgave 39	Flervalg
40	MAS234_2020h - Oppgave 40	Flervalg
41	MAS234_2020h - Oppgave 41	Flervalg

42	MAS234_2020h - Oppgave 42	Flervalg
43	MAS234_2020h - Oppgave 43	Flervalg
44	MAS234_2020h - Oppgave 44	Flervalg
45	MAS234_2020h - Oppgave 45	Flervalg
46	MAS234_2020h - Oppgave 46	Flervalg
47	MAS234_2020h - Oppgave 47	Flervalg
48	MAS234_2020h - Oppgave 48	Fyll inn tall
49	MAS234_2020h - Oppgave 49	Flervalg
50	MAS234_2020h - Oppgave 50	Flervalg
51	MAS234_2020h - Oppgave 51	Flervalg
52	MAS234_2020h - Oppgave 52	Flervalg
53	MAS234_2020h - Oppgave 53	Flervalg
54	MAS234_2020h - Oppgave 54	Flervalg

## 1 MAS234\_2020h - Oppgave 1

C++ er et *typef* språk. Dette vil si at:

**Velg ett alternativ:**

- Påstanden er usann.
- En variabel kan deklarereres med auto, og har da ikke en datatype.
- Det er vanskeligere for kompilatoren å finne feil.
- Det er ulovlig å konvertere mellom datatyper.
- En variabel har alltid en datatype.

## 2 MAS234\_2020h - Oppgave 2

Hva slags datatype er dette?

```
enum class Operation { predict=1, update=0 };
```

**Velg ett alternativ:**

- En enum med to mulige verdier, predict og update.
- Klasse, med to medlemsvariable.
- Dette er ikke en datatype.
- int (heltall).

### 3 MAS234\_2020h - Oppgave 3

Vi ønsker å deklarere et heltall uten fortegn, og med fast bredde 16 bit. Hvilken datatype velger du? Det kan antas at C++11 eller nyere er tilgjengelig.

Fyll inn ditt svar her:

### 4 MAS234\_2020h - Oppgave 4

Programmet ditt skal utføre en gjentagende oppgave 34 ganger. Antallet ønskede iterasjoner er altså kjent på forhånd.

Hvilken type løkke er mest hensiktsmessig for dette formålet?

**Velg ett alternativ:**

- do/while
- while
- loop
- for

## 5 MAS234\_2020h - Oppgave 5

Klassen KryptoWallet inneholder én medlemsvariabel og én public funksjon.

Funksjonen skal returnere konto-ID, og medlemsvariabelen inneholder den samme konto-ID.

Hvilket navn/identifikator er mest korrekt å velge til hver av dem?

**Finn de som passer sammen:**

	accountId_	accountId	_accountId_	Account_ID
Medlemsfunksjon	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Medlemsvariabel	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

## 6 MAS234\_2020h - Oppgave 6

Programmet skal skrive ut MAS234! til terminal. Tegnet \n på slutten gir linjeskift.

Fyll inn det som mangler for at dette enkle programmet skal kompile og kjøre! Husk å kvalifisere navnet.

---

```
#include <iostream>
```

```
int main()
{
    std::cout << "MAS234!\n";
    return 0;
}
```

## 7 MAS234\_2020h - Oppgave 7

Klasse og  struct  (enum, class enum, struct) er nesten det samme i C++. Den eneste signifikante forskjellen er default synlighet på funksjoner og variable.

## 8 MAS234\_2020h - Oppgave 8

Synlighet for variabler og funksjoner tilhørende en klasse styres med public, private og/eller protected.

Hva gjør private?

Ett av svarene under gir 2 poeng, ett av svarene gir 0 poeng, og ett av svarene gir -1 poeng.

**Velg ett alternativ:**

- Private øker sikkerheten, og forhindrer 100% at noen kan hente ut data fra minnet.
- Funksjoner og variable deklarert private kan kun aksesseres fra andre funksjoner i klassen selv.
- Variabler deklarert private kan alltid leses fra utsiden av klassen på samme måte som private-variabler.

## 9 MAS234\_2020h - Oppgave 9

Hva er den største verdien som kan legges inn i en 16 bit unsigned int?

Svar:  .

## 10 MAS234\_2020h - Oppgave 10

C++ er standardisert av ISO. Hvorfor er det viktig at språket er standardisert?

Det finnes mange gode grunner til å standardisere, men kun ett alternativene her er korrekt.

**Velg ett alternativ:**

- Programmer skrevet i standard C++ skal fungere med kompilatorer fra ulike leverandører.
- Det blir totalt sett billigere å ta frem en ny versjon av språket når mange deltar.
- Ingen deler av språket utvikler seg videre.
- Det går raskere å ta frem en ny versjon av språket.

## 11 MAS234\_2020h - Oppgave 11

I denne oppgaven ønsker vi å returnere to int-verdier fra en C++-funksjon.

Hvilken funksjons-deklarasjon er korrekt for dette formålet?

**Velg ett alternativ:**

- void calculate(int& v1, int & v2, const int input);
- int, int calculate(const int input);
- double calculate(const int input);
- int&, int& calculate(const int input);

**12 MAS234\_2020h - Oppgave 12**

```
#ifndef MYHEADER_H
#define MYHEADER_H

#include <cmath>

double unknownName(double a, double b) {
    return 1.0 + std::pow(a, b);
}

#endif // MYHEADER_H
```

Hva beregner funksjonen unknownName i illustrasjonen?

Ditt svar her:

Unngå mellomrom i svaret!

### 13 MAS234\_2020h - Oppgave 13

```
enum class Color { red, black };

bool verifyPercentage(std::vector<Color>& nonstop)
{
    int r = 0;
    int b = 0;
    for(auto& color : nonstop) {
        switch (color)
        {
            case Color::black:
                ++b;
                break;
            case Color::red:
                ++r;
                break;
        }
    }

    return r == b;
}
```

Bedriften Pepperkakehus AS har investert i en ny AI-algoritme for kvalitetskontroll.

AI-algoritmen detekterer nonstop og fargen på disse, og lager deretter en std::vector<Color>. Denne inneholder ett Color-element per detekerte nonstop.

Til slutt kalles funksjonen CakeControl::verifyPercentage, som skal sjekke at vi har **eksakt** 50% røde og 50% sorte nonstop på taket.

Implementasjonen av verifyPercentage er gitt i illustrasjonen.

**Velg ett alternativ:**

- Alt er perfekt, kjør på!
- enum class finnes ikke!
- Implementasjonen itererer ett hakk for langt, og ligger ikke i korrekt namespace.
- Implementasjonen er korrekt, men må legges i namespace CakeControl.

**14 MAS234\_2020h - Oppgave 14**

```
#include <iostream>

int main()
{
    char a[] = { 'M', 'A', 'S', '2', '3', '4', 'Y' };
    char* p1 = &a[4];
    char* p2 = &a[5];

    p2++;

    std::cout << *p1 << a[1] << a[2] << *p2 << std::endl;
}
```

Hva printes til terminal her? Se bort i fra linjeskift.

MERK: Case sensitivt svar. Vi skiller mellom små og store bokstaver her!

De fire tegnene som printes er: 3ASY

## 15 MAS234\_2020h - Oppgave 15

```
int main()
{
    const char c = 'a';

    switch(c)
    {
        case 'a':
            std::cout << "1 ";
        case 'b':
            std::cout << "2 ";
        default:
            std::cout << "3.";
    }

    return 0;
}
```

Programmet i illustrasjonen kompilerer, og kjøres.

Hva skrives ut til terminal?

**Velg ett alternativ:**

Ingen av disse.

1

1 2

1 2 3.

## 16 MAS234\_2020h - Oppgave 16

```
#include <iostream>

int main()
{
    const char m = 42;
    const char* a = &m;
    const char* s = a;

    std::cout << "Value is: " << ++(*s) << std::endl;
    return 0;
}
```

Hva skjer når vi forsøker å kompilere og kjøre programmet?

**Velg ett alternativ:**

- Programmet kompilerer ikke.
- Følgende printes til terminal: Value is 43
- Følgende printes til terminal: Value is 42
- Programmet kjører, men krasjer før cout.

## 17 MAS234\_2020h - Oppgave 17

Hvis vi oppretter en double-variabel som IKKE skal kunne endre verdi etter opprettelsen.

Hva må vi gjøre?

**Velg ett alternativ:**

- Skjule variabelen i et namespace.
- Deklarere variabelen const
- Lage en private variabel i en klasse.
- Deklarere variabelen static.
- Velge en enum.

**18 MAS234\_2020h - Oppgave 18**

```
#include <iostream>

int main()
{
    double measuredValues[5] {1.0, 5.0, -2.0, 8.3, 5.1};

    for (int i = 0; i <= 5; i++)
    {
        std::cout << measuredValues[i] << ", ";
    }
    return 0;
}
```

Hva skjer når vi forsøker å kompilere og kjøre eksempelet over?

**Velg ett alternativ:**

- Programmet kompilerer, men vil ikke starte.
- Programmet kompilerer ikke.
- Det kan vi ikke vite sikkert. Dette er et eksempel på udefinert oppførsel.
- De fem elementene i array-en printes ut etter hverandre.

**19 MAS234\_2020h - Oppgave 19**

Hva gjør følgende C++-kodesnutt?

```
PORTB &= ~ (1U << 3U) ;
```

**Velg ett alternativ:**

- Bit 3 på PORTB toggles, mens de andre IKKE påvirkes.
- Bit 3 på PORTB settes til lav, mens de andre bit-ene IKKE påvirkes.
- Dette er feil, U har ikke noe her å gjøre.
- Bit 4 på PORTB settes til lav, mens de andre bit-ene settes høye.

**20 MAS234\_2020h - Oppgave 20**

Hva gjør data direction-registerene på en AVR mikrokontroller?

**Velg ett alternativ:**

- Styrer intern pull-up-motstand.
- Styrer retningen på CAN-bus.
- Styrer retning på pinnene på en gitt port.
- Styrer UART, slik at vi kan bruke printf.

## **21 MAS234\_2020h - Oppgave 21**

Alle C++-funksjoner unntatt konstruktører og destruktører må ha returtype.

Hvilken returtype benyttes når vi IKKE ønsker å returnere noe?

Svar her:

## **22 MAS234\_2020h - Oppgave 22**

Funksjoner i C++ kan kalle seg selv. Dette heter rekursjon.

Hva er essensielt for en rekursiv funksjon?

**Velg ett alternativ:**

- Den må ikke ta inn parametre.
- Det må legges inn en stoppbetingelse, slik at den maksimalt kaller seg selv N ganger.
- Den må ikke ha returtype, slik at det unngås kopiering av data for hver runde.

## **23 MAS234\_2020h - Oppgave 23**

Vi kan ha mer enn én destruktur i en C++-klasse.

**Velg ett alternativ:**

- Sant
- Usant

## 24 MAS234\_2020h - Oppgave 24

Vi kan ha mer enn én konstruktur i en C++-klasse.

**Velg ett alternativ:**

Usant

Sant

## 25 MAS234\_2020h - Oppgave 25

Når skal destructor i C++ deklarereres virtual?

**Velg ett alternativ:**

Hvis vi eller andre kan tenkes å arve fra eller til klassen, nå eller senere, så skal den være virtual.

Bare hvis klassen er abstrakt.

Det er unødvendig -- også ved arv.

Det gjør programmet hakket raskere, men er ikke nødvendig. Hvis vi har store programmer, så bør destructor deklarereres virtual i de fleste tilfeller.

## 26 MAS234\_2020h - Oppgave 26

Hvor mange klasser kan vi arve fra i samme klasse i C++ ?

Altså; hvor mange "foreldreklasser" kan en klasse ha?

**Velg ett alternativ:**

- Maksimalt 2.
- Ingen grense av praktisk betydning (vi kan ha over 1000 foreldre-klasser).
- Maksimalt 10.
- Maksimalt 1.

## 27 MAS234\_2020h - Oppgave 27

Hvilken påstand er riktigst:

**Velg ett alternativ:**

- I C++ kalles alle funksjonene i Main-filen ved oppstart.
- Main-funksjonen er alltid startpunktet i et C++-program.
- Operativsystemet kaller main-funksjonen når et C++-program startes.
- Vi kan fritt velge navn på første funksjon som kalles i et C++-program.

**28 MAS234\_2020h - Oppgave 28**

Hvilket alternativ beskriver **IKKE** et tilfelle med behov for krysskompileering.

**Velg ett alternativ:**

- Kompilering av kildekode for iPad med ARM på en Intel x86 MS Windows-maskin.
- Kompilering av kildekode for embedded Linux på maskinen dette skal kjøres på (f.eks. Raspberry Pi).
- Kompilering av kildekode for mikrokontroller fra MacOS
- Kompilering av kildekode for Arduino på AMD64 PC med MS Windows.

**29 MAS234\_2020h - Oppgave 29**

Påstand: Innebygde datasystemer har lav ytelse pga. bruk av Arduino og mikrokontrollere.

**Velg ett alternativ:**

- Sant, mikrokontrollerene på Arduino er trege.
- Usant, det finnes mange godt egnede plattformer med høy ytelse, som ikke nødvendigvis er mikrokontrollerbasert. Mikrokontrollere kan også gi høy ytelse til en rekke formål.

**30 MAS234\_2020h - Oppgave 30**

Det skal utvikles styrings-computer for en drone på < 250 gram.

Hva vil du vektlegge? Pris er allerede med i vurderingen.

**Velg ett alternativ:**

- Høy ytelse og tilgjengelig IO.
- Lav vekt, tilgjengelig IO og tilstrekkelig ytelse.
- Lav vekt, lavt effektforbruk og høy ytelse.

**31 MAS234\_2020h - Oppgave 31**

Hva har GIT versjonskontroll til felles med kryptovalutaen Bitcoin?

**Velg ett alternativ:**

- Kryptosjekksummer benyttes for å sikre at ingen kan tukle med/endre historikk uten at det detekteres.
- Begge deler er shady.
- Utviklere er sikret anonymitet fordi historikk enkelt kan endres i ettertid. Det er viktig hvis f.eks. en privat epost-adresse ved en feil legges inn som kommentar i kildekode som sjekkes inn.
- Samme person står bak / har funnet opp begge deler, og GIT er avhengig av Bitcoin for å fungere.

## 32 MAS234\_2020h - Oppgave 32

Hva slags type versjonskontroll er GIT?

**Velg ett alternativ:**

- En webside
- Distribuert (potensielt)
- Lokal (unngå konflikter)
- Sentralisert (alltid)

## 33 MAS234\_2020h - Oppgave 33

Du skal hente et GIT-repository fra Github for å gjøre noen forbedringer. Hva er korrekt fremgangsmåte fra terminalen?

Anta at du står i rett folder, der du ønsker repository-et plassert.

Anta også at repository free-falling-cat eksisterer. (Det eksisterer imidlertid ikke i virkeligheten, så samtlige alternativer vil feile i praksis!).

**Velg ett alternativ:**

- git checkout https://github.com/uia-mechatronics/free-falling-cat.git
- git clone https://github.com/uia-mechatronics/free-falling-cat.git
- git rework https://github.com/uia-mechatronics/free-falling-cat.git
- git copy uia-mechatronics:free-falling-cat.git

## 34 MAS234\_2020h - Oppgave 34

Terminalen i f.eks. Ubuntu Linux er et kraftig verktøy, og kjøring av kommandoer dere ikke vet hva gjør, må unngås.

Dette er spesielt viktig dersom vi er innlogget som rotbruker (root) eller når vi kjører noe med superbruker-rettigheter (sudo).

Anta at du er innlogget som bruker root.

Hvilken av de følgende kommandoer er farligst? **ADVARSEL: IKKE PRØV KOMMANDOENE!**

**Velg ett alternativ:**

- rm -rf /
- touch /home/username/fil.txt
- fdisk -l
- wget http://some-random-website.net/CppClass.cpp

## 35 MAS234\_2020h - Oppgave 35

For å sjekke at overføring av data, f.eks. over en serielinje, har gått bra, kan vi benytte feildeteksjon.

Sortér de følgende typene feildeteksjon slik at nummer 1 er den med mulighet til på detektere flest typer overføringsfeil, og nummer 3 er den som detekterer færrest typer feil.

**Velg korrekt rekkefølge:**

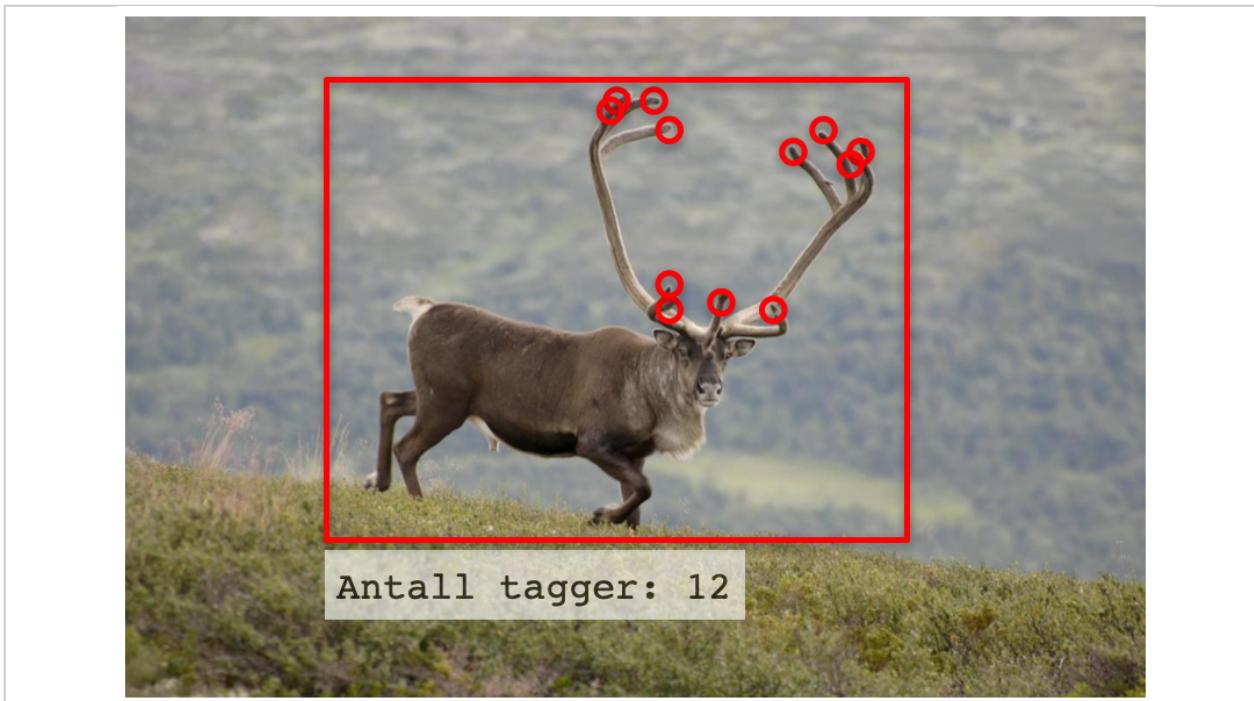
	2	1	3
Sjekksum (f.eks. SHA-256)	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Cyclic Redundancy Check (CRC)	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Paritetsbit	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>

**36 MAS234\_2020h - Oppgave 36**

Hva er en nettverkstopologi?

**Velg ett alternativ:**

- En nettverkstopologi gir oss adressene på enhetene i nettverket.
- En nettverkstopologi beskriver hvordan noder i nettverket er koblet sammen.
- En nettverkstopologi forteller hvilken type nettverk som benyttes (f.eks. CAN og Ethernet).
- En nettverkstopologi beskriver hvordan noder i nettverket er koblet sammen, for trådløse nettverk har vi derfor ingen topologi.

**37 MAS234\_2020h - Oppgave 37**

Du skal sjekke uvanlige feilmeldinger på en datamaskin som overvåker bestanden av reinsdyr på Hardangervidda.

Maskinen er tilkoblet internett med public IP, og du har satt opp denne maskinen selv.

Hva vil være det beste alternativet for fjerntilgang når maskinen er Linux-basert? (Det finnes andre alternativer, men velg det beste i listen under).

**Velg ett alternativ:**

- Remote Desktop (RDP)
- telnet
- UART-kabel
- SSH
- SSH som rotbruker
- telnet som rotbruker
- Remote Desktop (RDP) som domene-administrator

**38 MAS234\_2020h - Oppgave 38**

Hva bør vi minimum forvente fra et sanntidsoperativsystem, som eksempelvis BlackBerry QNX Neutrino og Microsoft Azure RTOS ?

**Velg ett alternativ:**

- Deterministisk oppførsel.
- Lavt effektforbruk.
- Støtte for svært mange ulike maskiner og arkitekturen.
- Høg gjennomsnittlig ytelse.

**39 MAS234\_2020h - Oppgave 39**

Hvilken av disse nettverkskonfigurasjonene har best sanntidsegenskaper i et multimaster-scenario:

**Velg ett alternativ:**

- USB med hub og 3 enheter.
- RS485 med 10 enheter på nettverket.
- Ethernet med >2 enheter på nettverket.
- CAN med 24 enheter på nettverket.

**40 MAS234\_2020h - Oppgave 40**

Hva er den beste beskrivelsen av bus-arbitrering:

**Velg ett alternativ:**

- Rekkefølgen nodene på bus-en starter i (slik at vi kan styre om dette er tilfeldig eller ikke).
- Fremgangsmåten for å få tilgang til delt kommunikasjonskanal.
- Antall ledere på bus-en, og hvordan hver av disse multiplekses.
- Monteringstype, f.eks. Wago-klemmer.

**41 MAS234\_2020h - Oppgave 41**

I konteksten operativsystemer. Hva beskriver en preemptive scheduler best?

**Velg ett alternativ:**

- En scheduler som sørger for færrest mulige avbrudd.
- En scheduler som sørger for høgest mulig gjennomsnittlig throughput.
- En scheduler som alltid når alle tidsfrister.
- En scheduler som kan avbryte oppgaver som kjører.

**42 MAS234\_2020h - Oppgave 42**

Hva er et eksempel på sanntidssystem av typen firm?

**Velg ett alternativ:**

- Sanntids-streaming av akselerasjonskommandoer til industrirobot.
- Sanntids-streaming av video hvor det ikke er så farlig at én video-frame droppes.
- Styringssystem for balansering av en kort invertert pendel.
- Styringssystem for liten og lett drone.
- Styringssystem for et åpent ustabilt jagerfly.

**43 MAS234\_2020h - Oppgave 43**

En av hovedforskjellene mellom de to bussene I2C og SPI er:

**Velg ett alternativ:**

- SPI er adressert, I2C er IKKE adressert.
- I2C er adressert, SPI har dedikerte linjer for å velge enhet master kommuniserer med.
- SPI er multi-master, I2C styrer hvem som er master med et eget signal.
- I2C er full duplex, det er IKKE SPI.

**44 MAS234\_2020h - Oppgave 44**

Hva karakteriserer en småsignaldiode?

**Velg ett alternativ:**

- Strøm kan kun gå i én retning av gangen.
- Strøm kan kun gå i én retning.
- Spenningen øker når strøm går gjennom en småsignaldiode.
- Dioden kan bli ødelagt hvis du setter opp en svært lav spennin i feil retning.

**45 MAS234\_2020h - Oppgave 45**

Hva er formålet med å plassere (raske) kondensatorer på supplyspenningen nær en mikrokontroller?

**Velg ett alternativ:**

- Muliggjøre høyere langvarig strømtrekk på mikrokontrollerens utganger.
- Unngå spenningsregulator i kretsen.
- Unngå unødvendig elektromagnetisk stråling og oppnå mest mulig stabil spennin på mikrokontrolleren.
- Unngå dioder nær mikrokontrolleren.

**46 MAS234\_2020h - Oppgave 46**

Du skal forsyne en mikrokontroller med 3.3 volt, og har kun et knappecellebatteri med 1.2 V i kretsen.

Det er ikke plass til andre typer batterier i dette nye og miniatyriserte måleinstrumentet som konstrueres.

Hvilken type spenningsregulator velger du for forsyning av mikrokontrolleren?

**Velg ett alternativ:**

- Buck-converter.
- Lineær spenningsregulator.
- Boost-converter.
- Lineær LDO (Low Drop Out) spenningsregulator.

## 47 MAS234\_2020h - Oppgave 47

### Electrical and Optical Characteristics ( $T_A = 25^\circ\text{C}$ )

Chip			Lens	Absolute Maximum Ratings				Electro-Optical-Data's at 20mA					Viewing Angle $2 \theta \frac{1}{2}$ (deg)
Emitted Colour	Peak Wavelength $\lambda_p$ (nm)	Dominant Wavelength $\lambda_p$ (nm)		$\Delta\lambda$ (nm)	Pd (mW)	If (mA)	Peak If (mA)	Forward Voltage Vf (V)	typ.	max	Luminous Intensity $I_v$ (mcd)	min.	typ.
Blue	470	470	water clear	-	123	35	110	3.2	3.5	4880	7300	9750	15°

\*Peak Forward Current (1/10 Duty Cycle, 10µs Pulse Width)

### Absolute Maximum Ratings ( $T_A = 25^\circ\text{C}$ )

Reverse Voltage	5V
Reverse Current ( $V_R = 5\text{V}$ )	$\leq 50\mu\text{A}$
Operating Temperature Range	-30°C ~ +85°C
Storage Temperature Range	-40°C ~ +100°C
Lead Soldering Temperature (3mm below the body)	265°C for 10 seconds

Illustrasjonen viser et utsnitt fra databladet til en lysdiode (LED).

LED-en skal kobles opp slik at den lyser konstant med maksimal effekt i henhold til databladet (uten bruk av PWM eller lignende).

Forsyningsspenningen er 12 V. Hvilken spesifikasjon velger du ved bestilling av den strømbegrensende motstanden?

**Velg ett alternativ:**

1/4 W, 220 ohm

1/4 W, 380 ohm

1/2 W, 120 ohm

1/2 W, 260 ohm

1/2 W, 251.48 ohm

**48 MAS234\_2020h - Oppgave 48**

```
uint8_t i = 0U;  
  
while(i < 5U) {  
    addItem();  
    i++;  
}  
  
std::cout << +i << std::endl;
```

Hvor mange runder går while-løkken i figuren?

Merk at du ikke får vite hva funksjonen addItem gjør, men det spiller ikke noen rolle her.

Programmet kompilerer og kjører uten problemer.

Svar: while-løkken kjører  runder før betingelsen blir false.

**49 MAS234\_2020h - Oppgave 49**

```
#include <vector>
#include <spotrobot.h>
#include <halodirobot.h>

int main()
{
    SpotRobot dawg;
    HalodiRobot eve;

    std::vector<ISayHello*> robots;

    robots.push_back(&dawg);
    robots.push_back(&eve);

    for (auto robot : robots)
    {
        robot->sayHello();
    }
}
```

Roboten Eve kan foreløpig kun hilse ved å vinke. Samtidig har MIL kjøpt in Boston Dynamics-roboten Spot. Den sistnevnte har dessverre ingen hale å hilse med, men den kan bjeffe.

Din oppgave er nå å lage et interface i C++ for å representere hilsing. Da kan både Eve-klassen og Spot-klassen arve dette interfacet, slik at vi kan få begge robotene til å hilse ved hjelp av en for-løkke..

Se et eksempel på et program som BRUKER dette interfacet i illustrasjonen.

Hva karakteriserer et interface (en abstrakt klasse) i C++ ?

**Velg ett alternativ:**

- Inneholder IKKE pure virtual destructor.
- Kan kun inneholde én funksjon (f.eks. for å hilse, bevege seg etc.).
- Inneholder MINST én virtual funksjon.
- Inneholder MINST én pure virtual funksjon.

**50 MAS234\_2020h - Oppgave 50**

```
#ifndef SPOTROBOT_H
#define SPOTROBOT_H

#include <isayhello.h>

class SpotRobot : [REDACTED]
{
public:
    SpotRobot() = default;
    ~SpotRobot() = default;

    // ISayHello interface
public:
    virtual void sayHello() override;
};

#endif // SPOTROBOT_H
```

Hva mangler her i klassen SpotRobot for å sikre at vi korrekt arver fra interfacet ISayHello. (Som altså er en abstrakt klasse).

Velg alternativet som skulle stått der den sorte firkanten sladder C++-koden.

**Velg ett alternativ:**

public ISayHello

inherit ISayHello

SayHello

private ISayHello

ISayHello

**51 MAS234\_2020h - Oppgave 51**

```
#ifndef SPOTROBOT_H
#define SPOTROBOT_H

#include <isayhello.h>

class SpotRobot : [REDACTED]
{
public:
    SpotRobot() = default;
    ~SpotRobot() = default;

    // ISayHello interface
public:
    virtual void sayHello() override;
};

#endif // SPOTROBOT_H
```

Hva er formålet med de to første og den siste linjen i header-filen vist i illustrasjonen?

**Velg ett alternativ:**

- Disse linjene er spesifikke for QT Creator, og har ikke noe med C++ å gjøre.
- Det er en markup som gjør at kompilatoren forstår at dette er en header-fil.
- Det er en type kommentar som gjør det tydelig hvilken fil dette er.
- "Include guard", sikrer at en header kun inkluderes én gang per kompileringsenhet.

**52 MAS234\_2020h - Oppgave 52**

Hva beskriver best et API i programmeringssammenheng?

**Velg ett alternativ:**

- Grensesnittet programmereren må forholde seg til ved bruk av f.eks. et bibliotek eller systemkall i operativsystemet.
- Hvordan data plasseres i minnet når data overføres ved funksjonskall.
- I Linux er det eneste API-et det som brukes når du kommuniserer med kjernen.
- Beskrivelsen av datatyper du kan overføre i et gitt grensesnitt.

**53 MAS234\_2020h - Oppgave 53**

```
#include <vector>
#include <spotrobot.h>
#include <halodirobot.h>

int main()
{
    SpotRobot dawg;
    HalodiRobot eve;

    std::vector<ISayHello*> robots;

    robots.push_back(&dawg);
    robots.push_back(&eve);

    for (auto robot : robots)
    {
        robot->sayHello();
    }
}
```

Hva gjør de følgende linjene i programmet vist i illustrasjonen?

```
robots.push_back(&dawg);
robots.push_back(&eve);
```

**Velg ett alternativ:**

- Erstatter elementene på de to første plassene i std::vector-en robots med henholdsvis dawg og eve.
- Legger inn en referanse til variablene dawg og eve i std::vector-en robots.
- Legger inn henholdsvis en peker til dawg og en peker til eve i std::vector-en robots.
- Legger inn en kopi av variablene dawg og eve i std::vector-en robots.

**54 MAS234\_2020h - Oppgave 54**

```
#ifndef SPOTROBOT_H
#define SPOTROBOT_H

#include <isayhello.h>

class SpotRobot : [REDACTED]
{
public:
    SpotRobot() = default;
    ~SpotRobot() = default;

    // ISayHello interface
public:
    virtual void sayHello() override;
};

#endif // SPOTROBOT_H
```

Hva betyr tilde-tegnet først i funksjonsnavnet ~SpotRobot ?

**Velg ett alternativ:**

- Invertering av returverdien.
- At funksjonen skal arves.
- Dette er en destruktur.
- Det er vanlig å bruke tilde ~ for å skille to funksjoner med samme navn fra hverandre.

**FAKULTET FOR TEKNOLOGI OG REALFAG**

**E K S A M E N**

**Emnekode: MAS234**

**Emnenavn: Innebygde datasystemer for mekatronikk**

Dato: 12. desember 2019

Varighet: 4 timer

Antall sider inkl. forside: 9

Tillatte hjelpe midler: Alle trykte og håndskrevne hjelpe midler tillatt. Kalkulator tillatt.

Merknader:

- Kandidaten må selv kontrollere at oppgavesettet er fullstendig.
- Les nøye igjennom oppgavene slik at du forstår hva det spørres etter.
- Beskriv eventuelle antagelser du må gjøre dersom oppgaven er formulert uklart. Lesing av spesifikasjoner, herunder oppgavetekster, er en del av det dere blir testet i.
- All kildekode i oppgaveteksten er gitt i C++11. Der det spørres om implementasjon i en eller annen form, skal det programmeres C++ på papir.
- Det kan antas at header <iostream> er inkludert fra før, alle andre nødvendige includes må spesifiseres i løsningen.
- Kodesnuttene i oppgaveteksten er i en del tilfeller listet ”frittstående”. Det skal antas at disse kjøres i kontekst av en funksjon (eksempelvis inni en main-funksjon).

---

## Oppgave 1 (10%)

Programforståelse.

Gitt følgende kodesnutt:

```
unsigned int x = 50;
unsigned int y = 100;

int a = x + y;

int* z = &a;
*z += 1;

std::cout << "a = " << a << ", z = " << *z << std::endl;
```

- Kodesnuten er plassert inni en main-funksjon i en cpp-fil. Hva skrives ut når programmet bygges og kjøres?
- Hvilken funksjon har & slik operatoren benyttes i kodesnutten?
- Kan variabelen a i kodesnutten deklarereres const? Hvorfor/hvorfor ikke?
- Hva er forskjellen på pre-inkrementering (eksempelvis ++i) og post-inkrementering (eksempelvis i++)? Gi et kort eksempel på bruk hvor oppførselen er forskjellig.

## Oppgave 2 (9%)

Funksjoner og løkker.

Det skal lages en funksjon som tar inn en array av double-elementer. Funksjonen skal returnere det største elementet.

- Skriv funksjonsdeklarasjonen.
- Lag implementasjonen til funksjonen fra deloppgave a).
- Hva vil være viktig her dersom funksjonen skal være effektiv på store arrays? Eksempelvis arrays med 100.000 elementer eller mer.

## Oppgave 3 (8%)

Standardbibliotek.

Det skal lages en funksjon som tar inn en `std::vector<double>`.

Funksjonen skal fjerne det siste elementet (merk at endringen også skal være synlig på utsiden av funksjonen!).

- a) Skriv funksjonsdeklarasjonen.
- b) Lag implementasjonen til funksjonen fra deloppgave a).

## Oppgave 4 (10%)

Programforståelse.

Gitt følgende:

```
class Something
{
    public:
        int x;
    private:
        int y;
};

int main()
{
    Something s;
    s.x = 314;
    s.y = 157;
    return 0;
}
```

- a) Hvorfor kompilerer ikke programmet?
- b) Nå skal klassen `Something` modifiseres slik at begge medlemsvariablene er deklarert `private`.

I tillegg skal det lages en funksjon som gir medlemsvariabelen `x` en verdi, og en funksjon som kan hente ut verdien til `x`. Kall funksjonene henholdsvis `setX` og `getX`.

Lag en ny deklarasjon til klassen som inkluderer endringene.

- c) Lag implementasjonen til `getX` og `setX`. Dette skal lages på samme måten som hvis det hadde vært plassert i en egen implementasjonsfil.
- d) Gi et eksempel på en vanlig konvensjon for filendelser på henholdsvis implementasjonsfiler og header-filer i C++.

## Oppgave 5 (15%)

Objektorientering.

Gitt følgende C++-klasser og main-funksjon.

En typisk bruk av filtrene vil være at vi har en array eller en std::vector hvor vi ønsker å kalle filter-funksjonen på hvert enkelt element i rekkefølge.

En annen bruk vil kunne være at vi leser en sensor med fast rate, f.eks. ti ganger i sekunder, og filtrerer verdiene vi leser inn før de brukes videre i programmet.

```
class Filter
{
public:
    virtual ~Filter() = default;
    virtual double filter(double value) = 0;
};

class NoPassFilter : public Filter
{
public:
    virtual ~NoPassFilter() = default;

    virtual double filter(double value) override
    {
        return 0.0;
    }
};

class PassThroughFilter : public Filter
{
public:
    virtual ~PassThroughFilter() = default;

    virtual double filter(double value) override
    {
        return value;
    }
};
```

// Oppgave 5 fortsettelse:

```
int main()
{
    Filter* filter = new NoPassFilter();

    const double unfilteredValue = 0.9;
    const double filteredValue = filter->filter(unfilteredValue);

    std::cout << "Unfiltered value is: " << unfilteredValue
        << ", filtered value is: " << filteredValue
        << std::endl;

    return 0;
}
```

- a) Hva skrives ut dersom programmet bygges og kjøres?
- b) Det skal byttes filter. Vi ønsker å benytte et PassThroughFilter i stedet for filteret som er benyttet nå. Vis endrede linjer.
- c) Lag en ny filterklasse som også arver fra Filter. Din nye filterklasse skal sende signalet rett igjennom («pass through») frem til første verdi lik 0.0 dukker opp, og fra og med dette tidspunktet skal filteret kun returnere 0.0.

Gi filterklassen et passende navn. Deklarasjon og implementasjon kan lages på samme sted, som i eksemplene over.

## Oppgave 6 (10%)

Systemmodellering.

- a) I SysML benyttes blant annet aktivitetsdiagram for å modellere systemers oppførsel.  
Hvilke diagramtyper er egnet for å modellere systemers struktur?
- b) Hva kjennetegner en god systemmodell i MBSE-sammenheng?

## Oppgave 7 (10%)

Begrepsforståelse.

- a) Hva er et namespace i C++ ?
- b) Lag et namespace med navn controller, og opprett en const-variabel av typen int inni namespace-et. Gi variabelen et valgfritt navn og verdi 2.
- c) Hva gjør preprosessoren når vi bygger et C++-program?
- d) Hva gjør kompilatoren når vi bygger et C++-program?

## Oppgave 8 (9%)

Kommunikasjon

- a) Hva menes med begrepet integritet i sammenhengen overføring av digitale data?
- b) Nevn to metoder som kan benyttes for å verifisere integritet på en overført digital melding. Hvilke typer feil kan detekteres?
- c) Nevn en metode som kan benyttes for å korrigere for feil i en overført melding. Hvilke typer feil kan korrigeres med denne metoden?

## Oppgave 9 (6%)

Nettverk.

Det skal overføres data mellom 3-5 enheter. To av enhetene er henholdsvis en motorstyringsnode og en node med hastighetsregulator.

Fysisk avstand mellom nodene er 200-3000 meter.

- a) Velg en passende type nettverk/kommunikasjonsstandard og begrunn valget.
- b) Hvilken topologi vil du benytte for oppkoblingen? Lag en enkel figur som illustrerer valget.

## Oppgave 10 (7%)

Operativsystemer.

Forklar kort:

- 1) Hva er den viktigste forskjellen på en tråd og en prosess?
- 2) Hva er prosessisolasjon (eng. «process isolation»)?

## Oppgave 11 (6%)

Single board computer vs. mikrokontroller. Operativsystem/ikke-operativsystem.

Velg typen embedded-maskin du mener er egnet for følgende to applikasjoner. Gi en kort begrunnelse for valget.

- a) Styring av signalgenerator/funksjonsgenerator (faktisk generere signalene). Det er behov for sinus-signaler med frekvens opp til 20 kHz.
- b) Logging av tidsserie-data fra solcellepanel-system (spenning, strøm, effekt, temperatur etc.), og tilgjengeliggjøring av loggede data på HMI (brukergrensesnitt) og skytjeneste på internett.

## FAKULTET FOR TEKNOLOGI OG REALFAG

## E K S A M E N

**Emnekode:** MAS234**Emnenavn:** Innebygde datasystemer for mekatronikk

Dato: 11. desember 2018

Varighet: 4 timer

Antall sider inkl. forside: 10

Tillatte hjelpe midler: Alle trykte og håndskrevne hjelpe midler tillatt. Kalkulator tillatt.

Merknader:

- Kandidaten må selv kontrollere at oppgavesettet er fullstendig.
- Les nøye igjennom oppgavene slik at du forstår hva det spørres etter.
- Beskriv eventuelle antagelser du må gjøre dersom oppgaven er formulert uklart. Lesing av spesifikasjoner, herunder oppgavetekster, er en del av det dere blir testet i.
- All kildekode i oppgaveteksten er gitt i C++11. Der det spørres om implementasjon i en eller annen form, skal det programmeres C++ på papir.
- Det kan antas at header <iostream> er inkludert fra før, alle andre nødvendige includes må spesifiseres i løsningen.
- Kodesnuttene i oppgaveteksten er i en del tilfeller listet ”frittstående”. Det skal antas at disse kjøres i kontekst av en funksjon (eksempelvis inni en main-funksjon).

---

## Oppgave 1 (7%)

Programforståelse.

Gitt følgende kodesnutt:

```
const int x = 3;
const int y = 2;
int z = 82;

const int a = x/y;
int* b = &z;

std::cout << "a = " << a << ", b = " << *b << std::endl;
```

- Kodesnuten er plassert inni en main-funksjon i en cpp-fil. Hvis vi antar at cpp-filen kompileres og programmet kjøres: Hva skrives ut?
- Hva betyr \* på linje 6 i kodesnuten?
- Hvilken funksjon har \* slik operatoren benyttes på siste linje i kodesnuten?
- Kodesnuten er plassert inni en main-funksjon og er opprinnelig laget for bruk på en PC med MS Windows 10. Du bestemmer deg for å skrive om programmet for bruk med Arduino. Hvor plasserer du kodesnuten nå, hvis den fremdeles skal kjøres én gang. Og må noe annet endres?

Det er ikke nødvendig å skrive om programmet for Arduino for å løse denne deloppgaven.

## Oppgave 2 (7%)

Funksjoner.

- Skriv deklarasjonen til en funksjon som tar inn to heltall av valgfri heltallstype og returnerer en bool. Funksjonen skal hete isGreater.
- Lag implementasjonen til funksjonen fra deloppgave a). Funksjonen skal returnere true hvis første argument er større enn andre argument, false ellers.

## Oppgave 3 (15%)

Løkker.

Følgende kodesnutt printer tall til standard output (vanligvis terminalvinduet).

```
const uint8_t n = 25;

for (uint8_t i = 0; i <= n; i++)
{
    std::cout << i << std::endl;
}
```

- a) Hva er første og siste tall som skrives ut?
- b) Hvor mange tall skrives ut?
- c) Hvorfor bør variabelen "n" være const i denne kodesnutten?
- d) Skriv en tilsvarende implementasjon som skriver ut hvert fjerde tall, fra og med 4, til og med 28.

## Oppgave 4 (8%)

Programforståelse.

Følgende funksjon er skrevet for å bytte verdi mellom to variabler.

```
void swap(int first, int second)
{
    const int temp = first;
    first = second;
    second = temp;
}

int main()
{
    int x = 314;
    int y = -2300;

    swap(x, y);

    std::cout << "x=" << x << " y=" << y << std::endl;
    return 0;
}
```

- a) Vil programmet kompile, og er det kjørbart? Hva printes ut hvis det kan kjøres?
- b) Fungerer programmet i henhold til spesifikasjonen (altså bytter det verdien på variablene)? Hvis ikke, foreslå en endring ved å skrive en ny og fungerende swap-funksjon.
- c) Bør variablene x og y i main deklarereres const? Hvorfor / hvorfor ikke?

## Oppgave 5 – Objektorientering (15%)

Gitt følgende C++-klasser:

```
class SensorData
{
public:
    SensorData(long t, double v)
        : timeStamp_us(t)
        , value(v)
    {
    }

    long timeStamp_us;
    double value;
};

class ISensorDataReceiver
{
public:
    virtual ~ISensorDataReceiver() { }
    virtual void setSensorData(SensorData& sd) = 0;
};

class TemperatureReceiver : public ISensorDataReceiver
{
public:
    TemperatureReceiver() { }

    virtual ~TemperatureReceiver() { }

    virtual void setSensorData(SensorData& sd)
    {
        if (timeStamp_us_ > sd.timeStamp_us)
        {
            timeStamp_us_ = sd.timeStamp_us;
            temperature_ = sd.value;
        }
    }

private:
    long timeStamp_us_;
    double temperature_;
};
```

Oppgave 5 - fortsettelse fra forrige side:

- a) Skriv implementasjonen til en main-funksjon som:
  - I. Først oppretter et SensorData-objekt med tidsstempel 12345 og måleverdi 38.9
  - II. .. deretter oppretter et TemperatureReceiver-objekt,
  - III. .. og til slutt sender inn det opprettede SensorData-objektet til TemperatureReciver-instansen ved hjelp av setSensorData-funksjonen.
- b) Hvorfor kan det ikke opprettes en instans av ISensorDataReceiver-klassen, og hva kaller vi denne typen klasser i C++ ?  

(ISensorDataReceiver er et interface, og dette lages i C++11 ved bruk av typen klasse det spørres om).
- c) Formålet med sjekken i setSensorData-funksjonen er å sikre at vi kun tar i mot målinger som er nyere enn den forrige vi mottok.

Det har sneket seg inn en liten feil i setSensorData-funksjonen. Hva er feilen?

## Oppgave 6 (8%)

Udefinert oppførsel.

- a) Hva er udefinert oppførsel, slik det er definert i C++-standarden?
- b) Hvorfor er det viktig å unngå å skrive programmer hvor udefinert oppførsel forekommer?
- c) Gi et eksempel på udefinert oppførsel i C++.

## Oppgave 7 (5%)

Datatypen struct.

- a) Lag en struct som inneholder feltene sensorId, timeOfValidity og measuredTemperature. Velg passende datatyper, og begrunn valgene kort.

Struct-en skal kun deklarereres.

- b) Hva skiller en C++-struct fra en C++-klasse?

## Oppgave 8 (10%)

Begrepsforståelse.

- a) Beskriv kort hva som skjer når vi kompilerer et C++-program.
- b) Hva skiller ”ordinær” kompilering fra krysskompilering?
- c) Under mikrokontrollerlab-en med AVR mikrokontrollere, ble det benyttet en Atmel ICE for skriving til flash-minnet på mikrokontrolleren (med mer).

Skjer skriving av flash-minnet på mikrokontrolleren før eller etter krysskompilering?

## Oppgave 9 (10%)

Mikrokontrollerkretser

- a) Hvilken funksjon har typisk en spenningsregulator i en mikrokontrollerkrets?
- b) Nevn minst to typer spenningsregulatorer, og forklar hva som skiller disse.
- c) En rød LED med spenningsdrop  $V_f = 2.3 \text{ V}$  skal styres av en mikrokontroller med suppliespenning  $V_s = 5 \text{ V}$ . Tegn et enkelt kretsskjema for hvordan dette bør kobles.

Det er ønskelig med maksimal lysstyrke, i og med at anvendelsen er utendørs.  
Komponentenes levetid er imidlertid også viktig.

Se vedlegg A for utsnitt fra LED-ens datablad.

- d) Hvis mikrokontrolleren i deloppgave c) er av AVR-type; hvilke steg er nødvendige for å kunne styre den programmatisk når du har koblet opp i henhold til kretsskjemaet ditt fra c) ? (Hva må du gjøre i f.eks. et C++-program for å styre LED-en)?

## Oppgave 10 (5%)

Nettverkstopologier.

Hvilke typer (én eller flere) nettverkstopologier er støttet med CAN? Tegn ett eller flere eksempler.

## Oppgave 11 (5%)

Operativsystemer I.

Forklar kort:

- 1) Hva de viktigste oppgavene til et operativsystem er,
- 2) hvilke egenskaper i operativsystemet som er mest sentrale når det skal anvendes som kjøremiljø for styring av tidskritiske fysiske systemer. (Eksempelvis styring av et fysisk system med rask dynamikk).

## Oppgave 12 (5%)

Operativsystemer II.

Foreslå et passende operativsystem for hvert av de følgende tilfellene. Gi en 1-2 setningers begrunnelse for hvert svar.

- a) Utvikling av desktop-applikasjon for registrering av måledata i et laboratorium.
- b) Lukket sløyfe-styring av en selvbalanserende robot. (Inkludert indre sløyfer.)



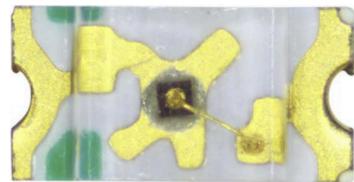
## Vedlegg A – Utsnitt fra datablad

## SMD LED

Red 2.1...2.6 V 0603

**SPECIFICATION:**

Package	0603
Light colour	Red
Transmission angle	130 °
Case colour	Transparent
Housing type	SMD
Conducting-state voltage max.	2.6 V
Conducting-state voltage min.	2.1 V
Dominant wavelength	631 nm
Forward current	30 mA
Forward voltage	2.1...2.6 V
Luminous intensity	80 mcd
Luminous intensity max.	80 mcd
Luminous intensity min.	80 mcd
Off-state current	10 µA
Off-state voltage	5 V
Operating temperature	-40...+85 °C
Peak wave length	645 nm
Power	78 mW
Technology	AlGaInP



## E K S A M E N

**Emnekode:** MAS234**Emnenavn:** Innebygde datasystemer for mekatronikk

Dato: 19. desember 2017

Varighet: 4 timer

Antall sider inkl. forside: 7

Tillatte hjelpe midler: Alle trykte og håndskrevne hjelpe midler tillatt. Kalkulator tillatt.

Merknader:

- Kandidaten må selv kontrollere at oppgavesettet er fullstendig.
- Les nøye igjennom oppgavene slik at du forstår hva det spørres etter.
- Beskriv eventuelle antagelser du må gjøre dersom oppgaven er formulert uklart. Lesing av spesifikasjoner, herunder oppgavetekster, er en del av det dere blir testet i.
- All kildekode i oppgaveteksten er gitt i C++11. Der det spørres om implementasjon i en eller annen form, skal det programmeres C++ på papir.
- Det kan antas at header <iostream> er inkludert fra før, alle andre nødvendige includes må spesifiseres i løsningen.
- Kodesnuttene i oppgaveteksten er i en del tilfeller listet ”frittstående”. Det skal antas at disse kjøres i kontekst av en funksjon (eksempelvis inni en main-funksjon).

---

## Oppgave 1 (7%)

Programforståelse.

Gitt følgende kodesnutt:

```
const int x = 3;
const int y = 2;
int z = 42;

const int a = x*y;
int& b = z;
b++;

std::cout << "a = " << a << ", b = " << b << std::endl;
```

- Kodesnuten er plassert inni en main-funksjon. Hva er spesielt med main-funksjonen i et C++-program?
- Hva skrives ut?
- Dersom kodesnuten utvides med følgende linje på slutten. Hva skrives ut i tillegg?

```
std::cout << "z = " << z << std::endl;
```

- Hva betyr & etter int på linje 6?

## Oppgave 2 (7%)

Funksjoner.

- Skriv deklarasjonen til en funksjon som tar inn en int og returnerer en bool. Funksjonen skal hete checkSubjectId.
- Lag implementasjonen til funksjonen fra deloppgave a). Funksjonen skal returnere true hvis tallet som sendes inn er 234, false ellers.

## Oppgave 3 (15%)

Løkker.

Følgende kodesnutt printer tall til standard output (vanligvis terminalvinduet).

```
const uint8_t n = 25;

for (uint8_t i = 0; i < n; ++i)
{
    std::cout << i << std::endl;
}
```

- a) Hva er første og siste tall som skrives ut?
- b) Hvor mange tall skrives ut, og skrives de ut etter hverandre horisontalt eller vertikalt?
- c) Kan variabelen "i" være const i denne kodesnutten?
- d) Skriv en tilsvarende implementasjon som skriver ut alle oddetall fra og med 5 til og med 131. Bruk den typen løkke som er best egnet for formålet.

## Oppgave 4 (8%)

Løkker.

Følgende kodesnutt er skrevet for å printe heltallene fra og med n til og med 0 i synkende rekkefølge til terminalen.

```
const uint8_t n = 10;

for (uint8_t i = n; i >= 0; --i)
{
    std::cout << i << std::endl;
}
```

- a) Er tallet 0 ett av tallene som printes til terminalen når programmet kjøres?
- b) Kodesnutten i denne oppgaven inneholder en rimelig ugrei feil. Hva er feilen, og hvordan kan denne rettes opp? (Flere ulike løsninger vil bli godtatt her).

Merk: Feilen gjelder ikke programsnuten i oppgave 3, selv om disse ligner noe strukturmessig.

## Oppgave 5 – Objektorientering (15%)

Gitt følgende C++-klasse:

```
class PidController : public IGenericController
{
public:
    PidController();
    virtual ~PidController();

    void setPid(double p, double i, double d);

private:
    void setP(double p);
    void setI(double i);
    void setD(double d);
};
```

- a) Skriv implementasjonen til en funksjon som oppretter en PidController-instans og setter regulatorparametrene til: P=4.2, I=0.5, D=0.0.
- b) Kan PidController-instansen være const? Hvorfor / hvorfor ikke?
- c) PidController arver fra IGenericController. IGenericController har en destructor som er deklarert virtual. Hva betyr dette?
- d) PidController arver fra IGenericController. IGenericController har en funksjon setTimeStepLength(double dt) som er deklarert pure virtual. Hva betyr dette for oss når vi lager klassen PidController?

Merk: Det er ikke listet kildekode for IGenericController, da denne ikke er nødvendig for å løse oppgaven.

## Oppgave 6 (10%)

Programforståelse.

En kryptert beskjed er mottatt og må dekrypteres. Følgende program utfører dekrypteringen:

```
void fun(char* message, unsigned int startIndex, unsigned
int stopIndex)
{
    for (unsigned int i = startIndex; i > stopIndex;)
    {
        std::cout << message[--i];
    }
    std::cout << std::endl;
}

void decryptMessage(char* message)
{
    const unsigned int christmasEve = 24;

    std::cout << "Navnet er: " << std::endl;

    fun(message, christmasEve/2 - 2, 0);
}

int main()
{
    char secretMessage[] = "naihsadraknu-gnoj";

    decryptMessage(secretMessage);

    return 0;
}
```

- Beskriv hva tredje parameter til fun-funksjonen gjør.
- Basert på innholdet i funksjonen "fun", foreslå et bedre og mer beskrivende navn på funksjonen. Navnet skal være skrevet C++-teknisk sett korrekt, og gjøre det unødvendig med kommentarer i koden for å forklare hva funksjonen gjør.
- Hva er den hemmelige beskjeden? Hint: En mye omtalt Kim.

## Oppgave 7 (4%)

Lag en enumerator som inneholder de mulige verdiene red, green og blue. Gi enumeratoren et beskrivende navn. Det skal *ikke* opprettes en instans av enumeratoren, den skal kun defineres.

## Oppgave 8 (10%)

Begrepsforståelse.

- Beskriv kort hva som skjer når vi kompilerer et C++-program.
- Hva skiller ”ordinær” kompilering fra krysskompilering?
- Hva mener vi med ”host” og ”target” når vi utvikler programvare for et innebygget datasystem? Tegn gjerne en enkel figur for å illustrere.

## Oppgave 9 (10%)

Mikrokontrollerkretser og CAN-bus

- Forklar kort hva en avkoblingskondensator er, og hvordan disse normalt bør plasseres relativt til f.eks. en mikrokontroller.
- Hva bør være foretrukket jordingsstrategi når vi designet et kretskort med flere relativt effektkrevende integrerte kretser (eller andre komponenter som har høyt strømtrekk)?

(Det samme gjelder også når vi designet en krets bestående av flere ”break-out-boards” etc..).

- Det er målt 100 kOhm motstand mellom Can High og Can Low på en CAN-bus. Hva kan dette indikere? Vil CAN-bus-en fungere i et slikt tilfelle?

## Oppgave 10 (7%)

Nettverkstopologier.

Tegn opp tre valgfrie vanlige nettverkstopologier og navngi disse.

## Oppgave 11 (7%)

Sanntidssystemer.

Velg de to punktene du mener beskriver et sanntidssystem best. (Skriv setningene fullt ut i besvarelsen).

- Et sanntidssystem må regne korrekt og ha svært god ytelse (kunne utføre mange beregninger raskt).
- Et sanntidssystem må forholde seg til tidsfrister.
- Et sanntidssystem må ha resultatet av beregningene klart til rett tid.
- Et sanntidssystem må har mange innganger og utganger.

# PRØVEEKSAMEN

**Emnekode:** MAS234

**Emnenavn:** Innebygde datasystemer for mekatronikk

Dato: **MAS234 PRØVEEKSAMEN 2017**

Varighet: Eksamens har 4 timers varighet, dette prøvesettet inneholder færre oppgaver enn en faktisk eksamen.

Antall sider inkl. forside: -

Tillatte hjelpeemidler: Alle trykte og håndskrevne hjelpeemidler tillatt. Kalkulator tillatt.

**Merknader:** Kandidaten må selv kontrollere at oppgavesettet er fullstendig.  
Les nøye igjennom oppgavene slik at du forstår hva det spørres etter. Beskriv eventuelle antagelser du må gjøre dersom oppgaven er formulert uklart. Lesing av spesifikasjoner, herunder oppgavetekster, er en del av det dere blir testet i. All kildekode i oppgaveteksten er gitt i C++11. Der det spørres om implementasjon, skal det programmeres på papir. Det kan antas at header <iostream> er inkludert fra før, alle andre nødvendige includes må spesifiseres i løsningen.

---

## Oppgave 1

Gitt følgende kodesnutt:

```
const int x = 3;
const int y = 2;
int z = 42;

const int a = x*y;
int& b = z;
b++;

std::cout << "a = " << a << ", b = " << b << std::endl;
```

- a) Hva skrives ut?
- b) Kan z være const i denne programsnuttens? Hvorfor / hvorfor ikke?

## Oppgave 2

Begrepsforståelse.

- a) Beskriv kort hva en "toolchain" er i forbindelse med krysskompilering. Nevn to av de viktigste komponentene i en toolchain.
- b) Vi har krysskompilert programmet vårt, og har fått ut en kjørbar fil. Vil denne kjørbare filen generelt sett kunne kjøres på host-maskinen? Hvorfor / hvorfor ikke?

## Oppgave 3

Arrays. Gitt følgende funksjon "arrayFun":

```
void arrayFun()
{
    int magNumbers[] = {1, 3, 7, 11, 24, 42};

    // a)
    std::cout << "a = " << magNumbers[2] << std::endl;

    // b)
    std::cout << "b = " << *magNumbers << std::endl;
}
```

- a) Hva skrives ut av linjen etter kommentar a) ?
- b) Hva skrives ut av linjen etter kommentar b) ?

## Oppgave 4

Funksjoner.

- a) Skriv deklarasjonen til en funksjon som tar inn to heltall av samme datatype og returnerer en bool. Funksjonen skal hete firstIsLargest. Velg datatype selv.
- b) Lag implementasjonen til funksjonen fra deloppgave a). Funksjonen skal returnere true hvis den første parameteren er størst, false hvis ikke.

## Oppgave 5

Bruk av SSH for fjerntilgang til innebygget datasystem.

Ved hjelp av en SSH-klient kan vi logge oss på en datamaskin som kjører eksempelvis Dropbear eller OpenSSH-server over et IP-nettverk. Dropbear ble benyttet under obligatorisk lab i MAS234.

- a) Hva bør sjekkes dersom følgende feilmelding dukker opp ved forsøk på tilkobling med SSH? Nevn de to-tre første sjekkpunktene dine i et slikt tilfelle.

**ssh: connect to host 10.0.0.3 port 22: Network is unreachable**

- b) Hvilke krav vil du sette til et passord for bruk til SSH-innlogging på et innebygget datasystem over internett? La oss si at maskinen styrer lastesystemet på et skip.

Hvis du ofte må logge inn på maskinen med SSH; hva kan gjøres for å få ned "arbeidsmengden" ved å taste et langt og sikkert passord – samtidig som graden av sikkerhet bevares eller økes? Svaret skal begrunnes.

## Oppgave 6

Løkker.

Følgende kodesnutt printer tall til standard out (vanligvis terminalvinduet).

```
const uint32_t n = 8;

for (uint8_t i = 5; i < n; ++i)
{
    std::cout << +i << std::endl;
}
```

- a) Hva er første og siste tall som skrives ut?
- b) Hvor mange tall skrives ut, og skrives de ut etter hverandre eller under hverandre?
- c) Hva gjør + foran i?

## Oppgave 7

Bruk av const.

- a) Hva betyr det dersom vi deklarerer en variabel const?
- b) Gi et kort eksempel (maksimalt åtte linjer) med bruk av const.
- c) Kompilerer det følgende programmet, hvis ikke – hvorfor?

Gi to eksempler hvor det med fordel kan benyttes const bør benytte const i et C++-program, og hva const gjør i disse tilfellene.

## Oppgave 8

Structs.

- a) Skriv deklarasjonen til en struct med navn "fluid" som inneholder tre felt: id, pricePerLitre og litre. Velg selv datatype på feltene.
- b) Lag en funksjon som tar inn en array med struct-typen du definerte i a) og summerer prisen for alle elementene. Funksjonen skal returnere prisen.

Hint: Prisen for ett element i arrayen er pricePerLitre ganget med antall liter.

Merk: Deloppgave b) kan løses selv om deloppgave a) ikke er besvart korrekt.

## Oppgave 9

Mikrokontrollerkretser – Krystall (med utdrag fra datablad). Tegn kretskjema for en AtMega128 mikrokontroller med strømforsyning og ekstern krystall.

Det er ikke nødvendig å tegne programmeringsgrensesnitt.

(Merk: jeg har ikke lagt ved datablad til prøveeksamen).