



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 hjelpemidler: Alle trykte og håndskrevne hjelpemidler 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;
```

- a) Kodesnutten er plassert inni en main-funksjon i en cpp-fil. Hvis vi antar at cpp-filen kompileres og programmet kjøres: Hva skrives ut?
- b) Hva betyr * på linje 6 i kodesnutten?
- c) Hvilken funksjon har * slik operatoren benyttes på siste linje i kodesnutten?
- d) Kodesnutten 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 kodesnutten 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.

- a) Skriv deklarasjonen til en funksjon som tar inn to heltall av valgfri heltallstype og returnerer en bool. Funksjonen skal hete isGreater.
- b) 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 kompilere, 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 deklarerer 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 TemperatureReceiver-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%)

Datotypen 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 spenningsdropp $V_f = 2.3 \text{ V}$ skal styres av en mikrokontroller med supplyspenning $V_s = 5 \text{ V}$. 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 selvb balanserende 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

