

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 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 (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;
```

- a) Kodesnutten er plassert inni en main-funksjon i en cpp-fil. Hva skrives ut når programmet bygges og kjøres?
- b) Hvilken funksjon har & slik operatoren benyttes i kodesnutten?
- c) Kan variabelen a i kodesnutten deklarerer const? Hvorfor/hvorfor ikke?
- d) 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.

- a) Skriv funksjonsdeklarasjonen.
- b) Lag implementasjonen til funksjonen fra deloppgave a).
- c) 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` av `double`-elementer. Funksjonen skal fjerne det siste elementet (merk at endringen også skal være synlig på utsiden av funksjonen!).

- Skriv funksjonsdeklarasjonen.
- 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;
}
```

- Hvorfor kompilerer ikke programmet?
- Nå skal klassen `Something` modifiseres slik at begge medlemsvariablene er deklartert `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.

