



Norges teknisk-naturvitenskapelige
universitet
Institutt for datateknikk og
informasjonsvitenskap

TDT4102 Prosedyre
og Objektorientert
programmering
Vår 2014

Øving 1

Frist: DD.MM.YYYY

Mål for denne øvinga:

- bli kjent med C++
- lære grunnleggende C++ og prosedyreorientert programmering
- lære hvordan programmer kan ta inn data og skrive ut på skjermen
- repetere grunnleggende programmering

Generelle krav:

- Bruk de eksakte navn og spesifikasjoner gitt i oppgaven.
- Det er valgfritt om du vil bruke en IDE (Visual Studio, XCode), men koden må være enkel å lese, kompilere og kjøre.

Anbefalt lesestoff:

- Kapittel 1 & 2, Absolute C++ (Walter Savitch)
- It's Learning notater

1 «Input/Output» og Funksjoner (15%)

- a) Skriv en funksjon `getAndPrintInteger` som lar brukeren skrive inn et heltall (integer), og skriver dette tallet ut på skjermen
Eksempel-resultat:
Skriv inn et tall: 42
Du skrev: 42
- b) Skriv en funksjon `getAndReturnInteger` som lar brukeren skrive inn et heltall (integer), og returnerer dette fra funksjonen, funksjonen skal *ikke* skrive tallet til skjerm
- c) Skriv en kommentar i koden din som forklarer hva slags returtyper disse funksjonene har, hvorfor de har dem, og forskjellen mellom disse returtypene. (minimum 5 setninger)
- d) Skriv en funksjon `getAndPrintSum` som ved å bruke en av funksjonene du nå har skrevet, tar inn to heltall og skriver ut kun summen på skjermen (*cout*).
- e) Skriv en kommentar i koden din som forklarer hvorfor du valgte å bruke den funksjonen du valgte (minimum 5 setninger)

2 Løkker (10%)

- a) Utvid funksjonen `getAndPrintSum` som du lagde i oppgave 1 til å la brukeren velge hvor mange tall som skal summeres, enten ved å angi antallet tall først, eller ved å slutte når brukeren gir tall med sum 0.
La den nye funksjonen hete `getAndPrintMoreSums`
- b) Skriv en kommentar i koden din som forklarer hvilken type løkke som er best egnet for hver av de to mulighetene i forrige oppgave, og hvorfor (minimum 5 setninger).
- c) Skriv funksjonen `getAndReturnDouble` som skal gjøre det samme som `getAndReturnInteger`, men istedenfor å lese inn et heltall, skal denne funksjonen lese inn et desimaltall
- d) Skriv en funksjon som konverterer NOK til Euro.
La brukeren gi beløpet som skal konverteres som et positivt desimaltall. Hvis brukeren gir et negativt tall skal programmet spør etter et nytt tall. Skriv ut det vekslende beløpet med to desimaler.
Bruk 1€ = 7,84 NOK.
Gjenbruk tidligere skrevne funksjoner så langt som mulig, og test funksjonen fra `main()`
- e) Skriv en kommentar i koden din som forklarer hvorfor vi ikke bør bruke `getAndReturnInteger` i forrige oppgave, men heller `getAndReturnDouble`, legg spesielt merke til hva i oppgaveteksten som legger føring på denne bruken. Kommenter også på returtypen til funksjonen du skrev i forrige oppgave. (Minimum 5 setninger).
- f) Hittil har vi hatt litt usystematisk testing av diverse funksjoner i `main`, dette skal vi nå rydde opp i.
Skriv om `main()` slik at brukeren kan velge i en meny mellom funksjonene fra foregående oppgaver, eksempel:

Velg funksjon:

- 0) Avslutt
- 1) Summer to tall
- 2) Summer flere tall
- 3) Konverter NOK til EURO.

Angi valg (0-3):

Hvis brukeren f.eks. velger 2, skal funksjonen som lar brukeren angi tall for summering kjøre, når denne er ferdig, skal menyen kjøre på nytt, programmet skal ikke avslutte før brukeren selv velger dette ved å angi menyvalget for "Avslutt".

- g) **Skriv en funksjon som skriver ut en gangetabell på skjermen (cout). La brukeren gi både bredde og høyde på tabellen.**

Velg selv navn for funksjonen

Hint: Løkke i løkke.

Legg denne funksjonen til i testmenyen

3 Flere operatorer (10%)

- a) **Skriv en funksjon som lar brukeren skrive inn et antall sekunder (ved å bruke *cin*) og skriver ut (til cout) tilsvarende timer, minutter og sekunder.**

Velg selv navn for funksjonen.

Eksempel-resultat:

Skriv antall sekunder: 10000

2 timer, 46 minutter og 40 sekunder

Programmet ditt trenger ikke være grammatisk korrekt, du kan med andre ord skrive ut "1 minutter" (Hvis du ønsker å gjøre programmet ditt grammatisk korrekt, kan du skrive ut entallbøyningen av *time*, *minutt* og *sekund* etterfulgt av "r" eller "er" *hvis* (if) antallet ikke er 1.)

Hint: Du kan bruke modulo operatoren (%) for å forenkle koden din. Modulo gir resten om man dividerer venstre side av uttrykket på høyre. For eksempel vil $72\%60$ gi 12

Legg denne funksjonen til i testmenyen fra forrige oppgave

- b) **Skriv en funksjon som regner ut mva og tips for en restaurantregning.**

- Måltidet koster f.eks. \$54.70 før mva og tips
- Mva er 8.75% av prisen (uten tips)
- Tips er 18% av prisen (inkludert mva)

La brukeren angi prisen, og test funksjonen med prisen angitt over. La programmet skrive ut prisen på måltidet, mva, tips og summen av alle tre på skjermen (cout).

Pass på at verdiene skrives ut med riktig presisjon (penger skal føres med 2 desimaler).

Legg denne funksjonen til i testmenyen

4 Kontrollstruktur (15%)

- a) **Skriv en funksjon som lar brukeren skrive inn et heltall og sjekker om tallet er et partall eller oddetall. La funksjonen skrive resultatet ut på skjermen (cout).**

Velg selv navn for funksjonen. Gjenbruk tidligere skrevne funksjoner hvis mulig.

Eksempel resultat:

Skriv inn et tall: 47

Tallet er et oddetall.

Legg denne funksjonen til i testmenyen

- b) **Skriv en funksjon som lar brukeren skrive inn to desimaltall og skriver ut hvilket som er størst på skjermen (cout).**

Velg selv navn for funksjonen. Gjenbruk tidligere skrevne funksjoner hvis mulig.

Eksempel resultat:

Skriv inn et tall: 3.14

Skriv inn et tall: 1.23

3.14 er størst

Legg denne funksjonen til i testmenyen

- c) **Skriv en kommentar i koden din hvor du forklarer hva som ville vært forskjellig dersom de foregående oppgavene hadde sagt «Skriv en funksjon som tar inn to desimaltall», istedenfor «Skriv en funksjon som lar brukeren skrive inn to desimaltall».**

(Minimum 5 setninger)

Å forstå forskjellen mellom disse to konseptene er viktig fremover i øvingsopplegget.

5 Bruk av funksjoner i funksjoner, og røtter (25%)

I funksjonene under skal du tolke hva som skal være argumenter til funksjonen, og lære å bruke funksjoner i funksjoner. Generelt heretter i faget, skal funksjoner returnere verdier, og ikke skrive ut noe til skjerm, med mindre annet er spesifisert.

- a) Skriv en funksjon `internalSum` som regner ut

$$b^2 - 4ac$$

og returnerer verdien (ingen utskrift til skjerm)

- b) Skriv en funksjon `positiveSqrt` som regner ut

$$\sqrt{x}$$

dersom `x` er positiv eller 0, i alle andre tilfeller skal funksjonen returnere -1.

Hint: Bruk en innebygd funksjon til å finne kvadratrot. Slå gjerne opp på <http://www.cplusplus.com/> for å finne ut hvilke funksjoner du har tilgjengelig.

- c) Skriv en funksjon `polyRoot` som bruker de to foregående funksjonene til å regne ut

$$\sqrt{b^2 - 4ac}$$

- d) Skriv en funksjon `abcFormula` som regner ut løsning(ene) til

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

der `a`, `b` og `c` er gitt som argumenter til funksjonen, gjenbruk funksjonene fra forrige oppgave, og skriv ut resultatene til skjerm.

Hint: Formelen har 2, 1, eller 0 reelle svar (vi ser bort ifra imaginære løsninger), dette bestemmes ved at:

2 svar hvis:

$$\sqrt{b^2 - 4ac} \neq 0$$

1 svar hvis:

$$\sqrt{b^2 - 4ac} = 0$$

0 svar hvis:

$$b^2 - 4ac < 0$$

Du har nå skrevet en funksjon for å løse andregradsligninger på formen

$$ax^2 + by + c = 0$$

- e) Lag en funksjon `solveAndPrintRoots` som lar brukeren skrive 3 desimaltall, bruk `abcFormula` til å regne ut røtten til andregradsuttrykket gitt ved disse tallene

- f) Legg til `solveAndPrintRoots` i `testMenyen` i `main()`

- g) Buk `testmenyen` til å regne ut røttene til andregradsuttrykket gitt ved disse tallene. Test programmet med verdier som gir 0, 1, og 2 løsninger

Hint: Prøv å regne ut røttene til:

$$x^2 + 2x + 4 = 0$$

$$4x^2 + 4x + 1 = 0$$

$$8x^2 + 4x - 1 = 0$$

NB: Generelt i øvingsopplegget bør dere gjøre noe tilsvarende dette for å teste at koden deres fungerer som den skal.

6 Flere Løkker (25%)

- a) Skriv en `calculateLoanPayments` funksjon som regner ut årlige innbetalinger av et lån over 10 år. La brukeren spesifisere lånebeløp og rente. La programmet for hver innbetaling skrive ut (cout) størrelsen på innbetalingen og det gjestående lånet. Utskriften skal være i et lettelleselig format.

Bruk følgende formel for å regne ut innbetalingene:

$$Innbetaling = \frac{Totalt_lån}{Antall_avdrag} + \frac{Rente}{100} * Gjestående_lån$$

Du trenger kun regne med én innbetaling i året, og denne skjer ved slutten av året. Renter skal ikke legges til eller trekkes fra det gjestående lånebeløpet, men kun regnes med i innbetalingen.

Legg denne funksjonen til i testmenyen

- b) Utvid funksjonen fra forrige deloppgave til å skrive ut oversikten over innbetalinger som en pen og pyntelig tabell, som ser ca slik ut:

År	Innbetaling	Gjestående Lån
1	100000	100000000
2	90000	1000000

Hint: Bruk `\t` som tabulator i utskriften