

Enkel inmatning och utmatning

Mål

I denna laboration repeterar du imperativ programmering i ljuset av programspråket C++. I första delen lär du dig hur formaterad inmatning och utmatning med de grundläggande datatyperna fungerar. Det är speciellt viktigt att förstå hur inmatningsbufferten fungerar. I andra delen övar du på att använda de vanliga styrstrukturerna för repetition och val.

Uppgift 1: Formaterad I/O

Läsanvisningar:

- Skriva, kompilera och köra huvudprogrammet `main`
- Inkludering av standardbibliotek (`#include`)
- Användning av standard namnrymd (`using namespace std`)
- Datatyper (`int`, `char`, `double`, `std::string`)
- Variabeldeklaration och initiering
- Formaterad inmatning (`cin`, `>>`, `ignore`, `getline`)
- Formaterad utmatning (`cout`, `<<`, `endl`, `setprecision`, `fixed`, `setw`, `setfill`)

Skriv ett program som hämtar data från tangentbordet enligt exemplen som följer. Det som syns på skärmen när man kör programmet skall stämma till fullo överens med det som står i exemplen. Detta gäller även antal blanktecken mellan ord och tal.

Ett tips kan vara att ni gör en liten del i taget och ser till att den fungerar för båda de givna körexemplen. På detta sätt undviker ni att få alla fel på en gång. Kommentera ut de sektioner som ni inte jobbar med för tillfällen så slipper ni skriva in hela inmatningen gång på gång.

KRAV: Du får i ditt program endast ha en variabel av respektive datatyp (det får alltså inte finnas t.ex. två stycken strängar). Det blir alltså bara fyra variabler i hela programmet.

KRAV: I denna deluppgift skall du endast använda rak sekvens, d.v.s. inga styrsatser av typen `if`, `while` eller `for`.

OBS! Ditt program skall klara av både körexempel 1 och 2. Dock behöver det inte klara av alla möjliga varianter av indata vilket i princip är omöjligt i detta läge.

Programmet ska ge resultat enligt följande exempel vid olika körningar (användarens indata markeras här med ***fet*** stil). Observera att eventuella inmatningar som är utöver de av programmet efterfrågade skall plockas bort innan nästa inmatning görs.

Körexempel 1

Skriv in ett heltal: **10**

Du skrev in talet: 10

Skriv in fem heltal: **12 30 27 13 11**

Du skrev in talen: 12, 30, 27, 13, 11

Skriv in ett flyttal: **15.20**

Du skrev in flyttalet: 15.200

Skriv in ett heltal och ett flyttal: **67 3.141592**

Du skrev in heltalet: 67

Du skrev in flyttalet: 3.142

Skriv in ett flyttal och ett heltal: **3.141592 54**

Du skrev in heltalet:-----54

Du skrev in flyttalet:----3.142

Skriv in ett tecken: **T**

Du skrev in tecknet: T

Skriv in en sträng: **Kalle**

Strängen 'Kalle' har 5 tecken.

Skriv in ett heltal och en sträng: **-5 Nisse**

Du skrev in talet |-5| och strängen |Nisse|.

Skriv in en sträng och ett flyttal: **Ria 1.2**

Du skrev in "1.200" och "Ria".

Skriv in en hel rad med text: **Bä, bä vita lamm har du**

Du skrev in textraden: 'Bä, bä vita lamm har du'

Skriv in en textrad som slutar med ett negativt heltal: **Det är kallt, -10**

Du skrev in textraden: 'Det är kallt, ' och heltalet '-10'

Körexempel 2

Skriv in ett heltal: **14.72**

Du skrev in talet: 14

Skriv in fem heltal: **10000 3860 27 -18** **0 0**

Du skrev in talen: 10000, 3860, 27, -18, 0

Skriv in ett flyttal: **15 Nisse**

Du skrev in flyttalet: 15.000

Skriv in ett heltal och ett flyttal: **7**

202.718 Dum text

Du skrev in heltalet: 7

Du skrev in flyttalet: 202.718

Skriv in ett flyttal och ett heltal: **3 54.435**

Du skrev in heltalet:-----54

Du skrev in flyttalet:----3.000

Skriv in ett tecken: **7**

Du skrev in tecknet: 7

Skriv in en sträng: **#4?".!**

Strängen '#4?".!' har 6 tecken.

Skriv in ett heltal och en sträng: **1 Ploja!**

Du skrev in talet |1| och strängen |Ploja!|.

Skriv in en sträng och ett flyttal: **Led**

012.3456.23

Du skrev in "12.346" och "Led".

Skriv in en hel rad med text: **Per är bra**

Du skrev in textraden: 'Per är bra'

Skriv in en textrad som slutar med ett negativt heltal:

23 -5

Du skrev in textraden: '

23 ' och heltalet '-5'

Uppgift 2: Momstabell

Läsanvisningar

- Styrord (if, for, while, do-while)
- Datatyper (int, float)
- Enkla funktioner med argument och returvärde
- Aritmetiska operatorer (+ - * / %)
- Jämförelse och logiska operatorer (< > <= >= == != ! && ||)
- Formaterad inläsning och utmatning (cin, cout, setfill, setw, left, right)

Konstruera ett program som skriver ut en momstabell. Programmet ska i terminalen fråga efter och sedan läsa in följande värden (där alla värden ska rimlighetskontrolleras)

- Nedre gräns för prisintervallet
- Övre gräns för prisintervallet
- Steglängd i tabellen
- Momsprocenten (uttryckt som decimaltal i intervallet 0 till 100 %)

Det som är viktigt i denna laboration är att användaren ska känna att den har stöd från programmet om den matar in felaktiga data (felaktiga i detta fall innebär att man matat in ett flyttal, men att inmatade värdet är orimligt i relation till vad programmet frågar efter). Körexempelen nedan visar hur programmet ska bete sig för just dessa indata. Det är givet att användaren *aldrig* kommer att mata in något data som är annat än flyttal.

KRAV: Du ska skriva en funktion för att beräkna momsen och en funktion för att beräkna priset inklusive moms. Båda funktionerna ska ta priset exklusive moms och momssatsen som parametrar och returnera momsen respektive priset inklusive moms.

KRAV: Du får bara använda variabler av heltalstyp samt datatypen **float**. Du får alltså *inte* använda datatypen **double**. Detta är för att du tydligare ska se de problem flyttal kan ge (som du råkar ut för även med datatypen double om du visar fler decimaler).

OBS! Du ska själv avgöra vad som är rimliga indata för respektive inmatning. Tanken är att användaren ska köpa saker från t.ex. en affär och att affären aldrig kommer att acceptera att den ska betala köparen något.

Programmet ska ge resultat enligt följande exempel vid olika körningar (användarens indata markeras här med ***fet*** stil).

Körexempel 1

INMATNINGSDEL

=====

Mata in första pris: **10.00**

Mata in sista pris: **15.00**

Mata in steglängd: **0.5**

Mata in momsprocent: **10.00**

MOMSTABELLEN

=====

Pris	Moms	Pris med moms

10.00	1.00	11.00
10.50	1.05	11.55
11.00	1.10	12.10
.	.	.
.	.	.
.	.	.
15.00	1.50	16.50

< Utelämnat av utrymmesskäl

Körexempel 2

INMATNINGSDEL

=====

Mata in första pris: **10.00**

Mata in sista pris: **12.00**

Mata in steglängd: **0.3**

Mata in momsprocent: **20.00**

MOMSTABELLEN

=====

Pris	Moms	Pris med moms

10.00	2.00	12.00
10.30	2.06	12.36
10.60	2.12	12.72
.	.	.
.	.	.
11.80	2.36	14.16

Körexempel 3

INMATNINGSDEL

=====

Mata in första pris: **-10.00**

FEL: Första pris måste vara minst 0 (noll) kronor

Mata in första pris: **100.00**

Mata in sista pris: **101.00**

Mata in steglängd: **-10.3**

FEL: steglängd måste vara minst 0.01

Mata in steglängd: **0.1**

Mata in momsprocent: **12.00**

MOMSTABELLEN

=====

Pris	Moms	Pris med moms

100.00	12.00	112.00
100.10	12.01	112.11
100.20	12.02	112.22
.	.	.
.	.	.
101.00	12.12	113.12

Sista "Pris utan moms"-värdet i andra exemplet är 11.80 och inte 12.00 (inte heller 12.10)! Detta beror på att det inte går jämnt upp med steglängden givet start- och slutpris. Inmatat "sista pris" ska då inte komma med i tabellen.

I tredje exemplet börjar användaren med att mata in ett felaktigt data. I denna uppgift är det meningen att ni ska fundera ut fler fall med orimliga indata och ge vettiga felmeddelanden för dem. Programmet ska köras tills användaren matar in korrekt data.

Utöver de givna exemplen på körning finns det många andra intressanta fall att testa. Nedan följer några förslag på värden. Kombinera lite olika för att se om ditt program verkar fungera.

Första pris: -1 0 10 100
Sista pris: -1 0 1 11 12 15 101 100000
Steg: -1 0 0.1 0.25 0.3 0.5 1 10
Momsprocent: -1 0 1 20 50 100 101