

Instruktion

Dette eksamenssæt indeholder 3 opgaver

Hver opgave vægter i den samlede karakter med de %-satser, der står ved hver af dem. De enkelte delopgaver indgår med forskellig vægt i den samlede vurdering.

Der er i opgaverne angivet en række header- og implementeringsfiler (.h, eller .cpp), der skal skrives for at løse opgaverne ved at skrive den relevante kode.

Alle disse header- og implementeringsfiler skal ved afleveringen **samles i én fil**, der afleveres **i PDF-format** på Digital Eksamen. Det skal fremgå i denne PDF-fil, hvilken opgave og fil de enkelte kode-dele kommer fra. Dvs. at kodefiler **ikke** skal afleveres enkeltvis eller som zip fil, ligesom Visual Studio solutions og projects **ikke** skal afleveres. **Al kode**, der skal bedømmes, **skal stå i denne ene fil**, der vil **ikke** blive kigget på kodefiler, der vedlægges som bilag.

Er der spørgsmål, der kræver et tekstsvar, noteres det i koden på det passende sted, evt. som en kommentar.

Der skal **ikke** inkluderes screen dumps af udskrifter fra programmet i afleveringen, men det er tilladt. I så fald kan de være i samme PDF fil, eller som bilag til afleveringen. Udskrifterne fra de kørende programmer vil først og fremmest være til fordel for dig, for at tjekke, at du har løst opgaven korrekt.

Der er ingen kodebilag til dette eksamenssæt.

Husk angivelse af eksamensnummer på første side.

Opgave 1 (35 %)

BMI betyder Body Mass Index og er et tal, der kan bruges til at give en indikation af, om man vejer for lidt, tilpas eller for meget.

Skriv et program, der kan udregne BMI (Body Mass Index) for indtastede værdier for højde i m og vægt i kg.

BMI udregnes som $\frac{vægt}{højde^2}$

- a) Bed brugeren om at indtaste vægt i kg og højde i m, indlæs dem og lav en udskrift, der udskriver den beregnede BMI. Brug kommatall (float eller double). Dit program behøver ikke at tage højde for om tallene er negative eller nul.
- b) Udvid programmet, så der udskrives en tekst, der angiver hvilken kategori personen med de indtastede data er i, ud fra følgende tabel.

BMI	Kategori
$\leq 18,5$	Undervægtig
$> 18,5$ og ≤ 25	Normalvægtig
> 25 og ≤ 30	Overvægtig
> 30	Svært overvægtig

Opgavesættet fortsætter på næste side.

Opgave 2 (20 %)

Lav en funktion med følgende prototype

```
void addSecondToFirst(int* f, int* s);
```

Den tager to integer pointers som input og den skal erstatte værdien som refereres af pointeren f med summen af de værdier der refereres af pointerne f og s

a) Implementer funktionen med den beskrevne prototype og opførsel

Funktionen skal testes. Dette skal gøres ved, at der skrives et program, hvor brugeren bliver bedt om at indtaste to heltal med et mellemrum imellem disse. Heltallene indlæses og programmet kalder ovenstående funktion på tallene med første tal som 1. parameter f og andet tal som anden parameter s og udskriver derefter indholdet af den variable der blev anvendt til første input.

b) Skriv main() funktionen til et program, der implementerer testen som beskrevet

Opgavesættet fortsætter på næste side.

Opgave 3 (45%)

De elektriske modstand for en leder er givet ved formelen

$$R = \rho L / A$$

hvor

R er modstanden [Ω]

L er længden af lederen [m]

A er tværsnitsarealet [m^2]

ρ (det græske bogstav *rho*) er den specifikke modstand for det specifikke materiale [$\Omega \cdot \text{m}$]

Eksempler er $\rho_{\text{guld}}: 0,0244 \cdot 10^{-6} \Omega \cdot \text{m}$

Eksempler er $\rho_{\text{rustfrit stål}}: 0,72 \cdot 10^{-6} \Omega \cdot \text{m}$

- a) **Definer i en h fil en klasse `Wire` der har private attributter for længde, tværsnitsareal og specifik modstand. Vælg selv datatype for disse attributter.**
- b) **Argumenter for valget af datatyper.**
- c) **definer yderligere i h filen:**
 - en **constructor** med parametrene specifik modstand, længde og tværsnitsareal
 - en metode `getModstand()`, som skal returnere den konkrete modstand for objektet. Lad metoden returnere den datatype du mener er korrekt og argumenter for dette.
- d) **Implementer klassen `Wire` i tilhørende cpp fil efter følgende specifikation:**
 - **Constructor** må ikke tillade nogen af parametrene at være negative eller 0. Hvis de er, skal den pågældende værdi sættes til 0 og en fejlbesked skal skrives til konsollen.
 - Metoden `getModstand()` skal implementeres i henhold til ovenstående formel. Hvis blot en af værdierne er 0 skal resultatet være 0

Opgaven fortsætter på næste side.
--

e) Lav en main funktion der tester klassen efter følgende specifikation:

- Lav tre objekter med negativt eller 0 input for hver af de 3 parametre, efter tur. Kald `getModstand()` for at teste dem.
- Lav test af metoden `getModstand()` med input: Længde: 1 m, Areal: 12 mm^2 ($12 \cdot 10^{-6} \text{ m}^2$) og de to typer materiale guld og rustfrit stål givet i opgaven. For guld skal dette give resultatet 0,002033 og for rustfrit stål skal resultatet være 0,06.

Opgaven fortsætter på næste side.
--

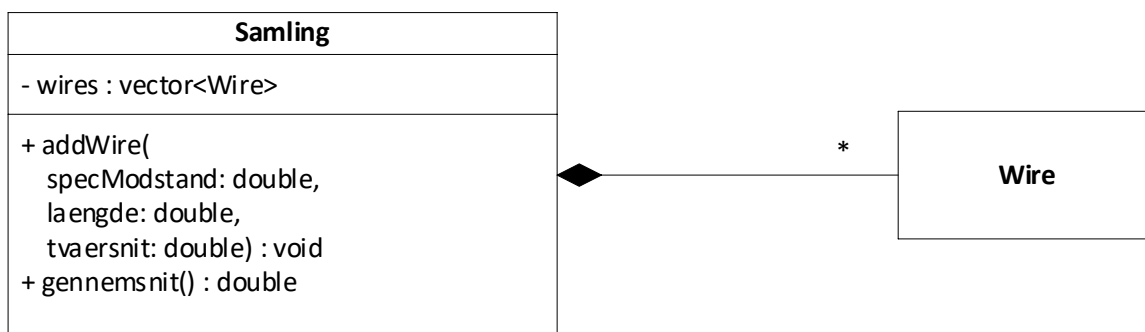
I de følgende delopgaver laves først en generel funktion for et antal objekter af klassen `Wire`. Dernæst skal der laves en klasse `Samling`, der administrer et antal objekter, og bruger den generelle funktion.

f) Implementer en funktion med følgende prototype:

```
double gennemsnitligModstand(vector<Wire> v);
```

som returnerer gennemsnittet af modstanden for alle `Wire` objekter i `vector`'en `v`.

Implementer følgende ekstra klasse (`Samling`). `Samling` skal indeholde en liste af `Wire` objekter implementeret ved hjælp af `vector` og to metoder `addWire` og `gennemsnit` (se diagram nedenfor).



g) Lav klasse deklARATIONEN i en header fil `Samling.h`.

h) Implementer klassen `Samling` i en tilhørende `cpp` fil. Metoden `addWire()` skal oprette et `Wire` objekt og tilføje dette til `vector`'en `wires`. Metoden `gennemsnit()` returnerer gennemsnittet af modstanden for enkelte `Wire` objekter i `vector`'en `wires`.

i) Test ved tilføjelse af mere kode til `main()` som bruger de to `Wire` objekter oprettet tidligere til indsættelse i et `Samling` objekt (gennemsnittet skal være `0,0310167`).

Bemærk at en `double/float` $0,72 \cdot 10^{-6}$ i C++ skrives som `0.72E-6`