

Programmieren in C++ – Sommersemester 2015

## Aufgabenblatt 2

**Abgabe: Donnerstag, 21.05.2015, 11:30**

### **Aufgabe 1 (Funktionsüberladung, Vorgabeargumente):**

- a) Definieren Sie eine Funktion **square**, die für eine Gleitpunktzahl ihr Quadrat zurückgibt. Definieren Sie anschließend eine Struktur **complex** mit den Elementen Real- und Imaginärteil, und definieren Sie die Funktion **square** auch für ein Element der Struktur **complex**. Geben Sie im Hauptprogramm beispielhaft eine Gleitpunktzahl sowie eine komplexe Zahl aus und jeweils ihr Quadrat, berechnet mit der Funktion **square**.
- b) Schreiben Sie ein Programm, das den Netto-Monatslohn aus den gearbeiteten Stunden, dem Stundenlohn und den zu zahlenden Steuern berechnet. Lassen Sie den Benutzer dazu die Anzahl der gearbeiteten Stunden eingeben. Fragen Sie den Benutzer danach, ob Stundenlohn und Steuersatz den Standardwerten entsprechen (7.50 Euro pro Stunde, 33%). Falls nicht, sollen die Werte vom Benutzer eingegeben werden. Schreiben Sie eine Funktion **gehalt** mit Vorgabeargumenten, die den Nettolohn aus den jeweiligen Werten für gearbeitete Stunden, Stundenlohn und Steuersatz berechnet; trennen Sie dabei Deklaration und Definition der Funktion. Rufen Sie die Funktion **gehalt** im Fall mit Standardwerten mit den Vorgabeargumenten auf, im allgemeinen Fall ohne die Vorgabeargumente.

Die Lösung beider Teilaufgaben soll sowohl die Implementierung als auch die Bildschirmausgabe des Programms umfassen.

## Aufgabe 2 (Funktionen mit Vorgabeargumenten, Funktionstemplates):

- a) Schreiben Sie eine Funktion **trapezFlaeche(...)** mit Vorgabeargumenten, die die Fläche eines Trapezes berechnet und auf dem Bildschirm ausgibt. Die Parameter der Funktion sollen die Länge der beiden parallelen Seiten, die Länge der Seite b und den Winkel Gamma angeben (siehe Graphik). Schreiben Sie die Funktion so, daß Sie für ein Parallelogramm nur die Seitenlängen von a und b und den Winkel Gamma, für ein Rechteck nur die Seitenlängen a und b sowie für ein Quadrat nur die Seitenlänge der Seite a angeben müssen.

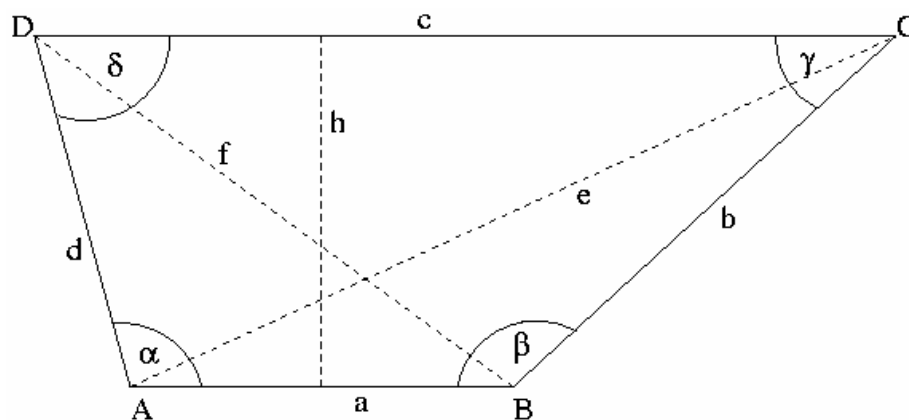
Testen Sie das Programm mit den folgenden Werten:

- i)  $a = 2.0$ ,  $c = 3.5$ ,  $b = 4.0$ ,  $\text{Gamma} = 45.0^\circ$
- ii)  $a = 2.0$ ,  $b = 4.0$ ,  $\text{Gamma} = 45^\circ$
- iii)  $a = 2.0$ ,  $b = 4.0$
- iv)  $a = 2.0$

Die Lösung der Aufgabe soll die Implementierung und die Bildschirmausgabe des Programms umfassen.

Hinweis: Die Funktion  $\sin(\cdot)$  findet sich in **<cmath>**.

Trapez:



Formeln zum Trapez	
Flächeninhalt	$A = \frac{1}{2} \cdot (a + c) \cdot h$
Höhe	$h = b \cdot \sin \gamma = d \cdot \sin \delta$
Hilfsgröße	$s = \frac{a + b + c + d}{2}$
Höhe	$h = \frac{2}{a - c} \sqrt{s(s - a + c)(s - b)(s - d)}$
Diagonalenlänge	$e = \sqrt{a^2 + b^2 - 2ab \cos \beta} = \sqrt{c^2 + d^2 - 2cd \cos \delta}$
Diagonalenlänge	$f = \sqrt{a^2 + d^2 - 2ad \cos \alpha} = \sqrt{b^2 + c^2 - 2bc \cos \gamma}$
Seitenlängen	$a, b, c, d$
Größen der Innenwinkel	$\alpha, \beta, \gamma, \delta$

(Quelle: HS Esslingen)

- b) Schreiben Sie ein Funktionstemplate für die Suche nach einem vorgegebenen Element in einem Feld eines beliebigen Datentyps. Die Funktion hat als Parameter das Feld und das zu suchende Element (sowie die Größe des Feldes). Der Rückgabewert der Funktion soll die Position des zu suchenden Elementes im Feld sein (erste Position also 1). Falls das Element nicht gefunden wird, soll der Rückgabewert (-1) sein. Es können C++-Standard-Datentypen angenommen werden.

Schreiben Sie weiterhin ein Funktionstemplate, das den Mittelwert aller Elemente des Feldes berechnet (Parameter: Feld und Länge des Feldes).

Testen Sie beide Funktionen mit Feldern vom Typ **int**, **long** und **double**.

Die Lösung der Aufgabe soll die Implementierung und die Bildschirmausgabe des Programms umfassen.

### Aufgabe 3 (Dynamische Speicherverwaltung):

Schreiben Sie ein Programm, das dynamisch Speicherplatz für  $n$  double-Zahlen erzeugt (wobei  $n$  vom Benutzer erfragt wird), die  $n$  Zahlen vom Benutzer eingeben lässt und die Summe, das Maximum und die Position des Maximums bestimmt und ausgibt,

- in C,
- in C++.

(Quelle: Uni Regensburg)