Aufgabe 1.3

Typen in C++

Typen:

bool: nimmt die Werte wahr oder falsch, bzw 1 oder 0 an.
Beispiel: bool b = false;
Zeichentyp(char):speichert einzelne Zeichen, unsigned char = vorzeichenlose Variante, signed char = Vorzeichen behaftete Variante, 8 Bit
Beispiel: char c = ‚a‘;
ganz zahlige Typen (int, short und long): speichern von ganzen Zahlen in verschiedenen Wertebereichen; mit unsigned = Vorzeichenlos, ohne Angabe oder mit signed = vorzeichenbehaftet; int 16 oder 32 Bit, short 16 Bit, long 32 Bit
Beispiel: int a = 9;
Fließkommatypen(float,double und long double): Speicherung von Fließkommazahlen mit verschiedener Genauigkeit, double = doppelte Genauigkeit
Beispiel: double d = 1.3;

Variablen:

- Ablegung von Werten auf die das Programm später zugreifen kann, können verändert werden
- Besitzen Namen und Typ z.b int, lokal oder global deklariert
- Lokale V gelöscht sobald sie Bezugsrahmen verlassen
- Globale V deklaration im Quelltext außerhalb von allen Funktionen

Werte:

- Zuweisung von Typen, z.b Fließkommazahlen, Zeichen, Ganze Zahlen, Wahrheitswerte

Was bedeutet const?:

- Const = Konstante, Behandlung wie Variable
- Wertzuweisung bei Deklaration, teilt Compiler mit dass ein Objekt oder Variable unveränderbar ist

Was ist eine Typkovertierung und warum kann es dabei Probleme geben?
Wenn einer Variable ein anderer Typ zugewiesen wird, z.B aus Float ein int. Probleme gibt es bei der Divison von Ganzzahlen weil immer eine Ganzzahl rauskommt auch wenn der Variablen eine Fließkomma zahl zugewiesen wird.

Aufgabe 1.4

Initialisierung:
Ist die Wert Zuweisung z.b int a = 7;.

Zuweisung:

- Nicht möglich wenn eine Variable const ist
- Oder der Typ den Zuweisungsoperator nicht unterstützt

Int a;
a = 7;

Aufgabe 1.5

Definition:

- Legt Namen & Typ der Variable fest, reserviert Speicherplatz

Deklaration:

- Legt Name & Typ der Variable fest, erst Typ dann Name
- Es können in dergleichen Anwendung mehrere Variablen hintereinander mit Komma getrennt deklariert werden, z.B int a, b c;

Schlüsselwörter 1.13

C++:

- von ISO genormte Programmiersprache, Weiterentwicklung von C

Quellcode:

- Beschreibung des Programmablaufs in Syntax der jew. Programmiersprache
- Festlegung wie sich Programm gegenüber von Nutzer, anderen Programmen verhält, welche Daten eingegeben werden können und wie sie verarbeitet und ausgegeben werden

Compiler:

- Übersetzung des Quellcodes in ausführbare Maschinensprache (für Computer verständl)

Linker:

- Computerprogramm welches einzelne Programmmodule zu einem ausführbaren Programm zusammenstellt
- Linkvorgang erfolgt nach Compilierung, letzter Arbeitsschritt bei Programmerstellung
- Statisches und dynamisches Linken

Objektcode:

- Zwischenergebnis des Compilevorgangs
- Bestehend aus Maschinencode für Architektur für die das Programm übersetzt wurde
- Enthält: kompakten & vorgeparsten Code, oft benutze Programmbibliotheken, die dann mit anderen Objetktdateien gebunden werden
- Format abhängig von Programmiersprache & Compiler, nach dem erstellen, erfolgt Linken, danach fertiges Programm

Ausführbare Datei:

- Kann als Computerprogramm ausgeführt werden, Binärdatei oder Maschinensprache oder Byte-Code, oder Textdatei die von Betriebssystem-Shell interpretiert wird

main():

- Hauptprogramm
- Als erstes beim Programmstart ausgeführt
- Ein oder zwei Parameter

#include:

- Integrieren Standertbibliotheken wie iostream & string

Kommentar:
// für einzeilige Kommentare oder /* */ für ganze Blöcke, Erklärungen von Funktionen oder Ergänzungen damit man den Quellcode leichter bzw zu einem späteren Zeitpunkt noch nachvollziehen kann

Header:

- C++ Dateien, die Funktionsdeklarationen und ähnliches enthalten
- Compiler weiß nicht was die Funktion tut kann sie aber aufrufen

Programm:

Ausgabe:

std::cout :

- Standart Ausgabe
- In iostream zur Verfügung gestellt

std::cin :

- Cin = console input
- Eingabe, in iostream zur Verfügung gestellt

<<:

- Ausgabeoperator

>>:

- Eingabeoperator

Funktion:

- Unterprogramm
- Sinnvoll für oft wiederholende Befehle
- Kann Parameter besitzen und Werte zurückgeben
- Muss deklariert sein, Eingabe und Rückgabewert, sonst void

Funktionssignatur:

- Name der Funktion, Anzahl und Reihnfolge der zugeweisungskompatiblen Parameterdatentypen

Gültigkeitsbereicht:

- Variablen die innerhalb der Funktion als static deklariert wurden, nur innerhalb der jeweiligen Funktion sichtbar, über Referenz oder Zeiger zugriff von außen möglich, Initialisierung erfolgt beim ersten Aufruf