Programmiersprachen
Übung 1

1.3)

Typen	**Variablen**	**Werte**
Int	a	9
Bool	b	false
Char	c	'a'
Double	d	1.3
Int const	two	2

Bei einer Änderung der Datendeklaration gibt **const** an, ob das Objekt oder die Variable nicht änderbar ist.

Mithilfe der Typkonvertierung kann ein Typ in einen anderen Typ konvertiert werden.
So kann ein int in ein double konvertiert werden.
Dabei wird immer in den höheren Typ konvertiert.
(z.B. short->int->long oder int->double)

Dabei kann es auch zu Problemen kommen, wird ein double in ein float konvertiert ist zum Beispiel die darausfolgende "Rundung" ein Problem oder das Konvertieren vom "höheren" in einen "niederen" Integertyp. (Ergebnisverfälschung)

1.4)

Initialisierung:

Mittels der Initialisierung wird die Variable auf einen initialen Anfangswert gesetzt, da sie durch die reine Deklaration und Definition einen beliebigen Wert hat.

```
int a=1;
int b=2;
```

Zuweisung:

Bei der Zuweisung wird der Wert des zweiten Operanden in dem durch den ersten Operanden angegebenen Objekt gespeichert. Im Gegensatz zu der Initialisierung kann ich die Werte von Variablen nach der Initialisierung durch Zuweisung ändern.

```
a = b;  // a=2
```

1.5)

Deklaration:

Durch die Deklaration wird eine Variable benannt und so dem Compiler bekannt.

Definition:

Bei der Definition wird einer Variable ein Speicherbereich zugeteilt, welcher eine eindeutige Adresse hat. Dieser wird dazu verwendet um Werte abspeichern zu können.

Int zahl;

1.6)

Die Signatur einer Methode/Funktion beinhaltet den Namen der Methode/Funktion und die Typen in ihrer Parameterliste. Dies ist besonders wichtig, wenn es mehrere Versionen von einer Methode/Funktion mit gleichem Namen, aber unterschiedlichen Parametern gibt.

Signatur:

sum(double a, double b)
square(int vat)

```
double sum(double a, double b)
{
        return a+b;
}
```

Gültigkeitsbereich von a & b nur innerhalb der Methode double sum.

```
int square (int var )
{
         return var * var;
}
```

Gültigkeitsbereich von var nur innerhalb der Methode int square.

```
int main ()
{
for ( int i = 0; i != 100; ++i)
        {
                std :: cout << "i^2 = " << square (i) << std :: endl ;
                std :: cout << "i+i = " << sum(i,i) << std :: endl ;
        }
return 0;
}
```

Gültigkeitsbereich von i nur innerhalb der Methode int main.
Bzw. Die Variable i wird jeweils a , b und var zugewiesen(Siehe Code).