# Vektoren und Pointer

Pointer sind und werde mit * gekennzeichnet

## Vertausche:

```
#include <stdio.h>

void vertauschen(int*,int*);

void main()
{
	int a= 5;
	int b= 7;

	printf("a= %i b= %i",a,b);
	vertauschen(&a,&b);

	printf("a=%i b=%i", a,b);

}

void vertauschen(int*a, int*b)
{
	int help;

	help = *a;
	*a = *b;
	*b = help;

	printf("a=%i b=%i",*a,*b);

}
```

## Pointer erstellen

```
#include <stdio.h>


void main()
{
	int i = 5;
	int *p = &i;

	printf("%i \n %i \n", i,*p);

	// &i Zeigt die Adresse und *p zeigt den Wert der in
	// i steht und p ohne * würde auch der Wert von i angezeigt werden
	printf("%i \n %i \n", &i, *p);

	// %p  für Pointer giebt die Hexa Adresse aus die die Pointervariable p hat
```

```
	printf(" %p \n %p \n",&i,p);
}
```

## Variablen Feldern

Int A[5]  bedeutet es wurden 5 Variablen  A aber mit INDEX  angelegt

Also:

```
A[0] = 0;
A[1] = 1;
A[2] = 2;
A[3] = 3;
A[4]= 4;
```

## Anwendung:

```
#include <stdio.h>

void main()
{
	int A[5];
	int i;

	for (i=0;i<5;i++)
		{
			A[i] = i;
			printf ("%i",A[i]);
		}
}
```

## Es werden 4 Eingaben gemacht und diese dann wieder ausgegeben

```
#include <stdio.h>

void Eingeben(int*);
void Ausgeben(int*);

void main ()
{
		int aiTest[5];
		Eingeben(aiTest);
		Ausgeben(aiTest);
}

			void Eingeben(int*p)
			{
				int i;

				for (i=0; i<5; i++)
				{
					scanf("%i", (p+i));
				}
			}


			void Ausgeben(int*p)
			{
				int i;
				for (i=0; i<5; i++)
				{
					printf("%i",p[i]);


					printf("%i",*(p+i));
				}
			}
```

## Aufgabe Summe 2ér Zahlen mit Array´s

```
#include <stdio.h>

void Eingabe(int*);
void Ausgabe(int*);

void main ()
{
	int Zahlen[2];

	Eingabe(Zahlen);
	Ausgabe(Zahlen);
}

		void Eingabe(int*Array)
		{
			printf("Zahl1 eingeben:\n");
			scanf("%i",Array);

			printf("Zahl2 eingeben:\n");
			scanf("%i",Array+1);
		}

	void Ausgabe(int*Array)
	{
		printf("\n %i und %i ist gleich %i \n",*(Array),*(Array+1),*(Array) + *(Array+1));
	}
```

## Durchschnitt

```
#include <stdio.h>

#define MAX 5


void lesen(int *);
float Durchschnitt (int *);


        void main()
        {
                int Vorgang[MAX];

                lesen(Vorgang);

                printf("Durchschnitt: %f\n", Durchschnitt(Vorgang));
        }


                        void lesen(int *Zaehler)
                        {

                                int i;

                                for (i = 0; i < MAX ; i++)
                                {
                                printf ("Gebe Zahl ein");
                                scanf ("%i", &Zaehler[i]);
                                }
                        }


                                                float Durchschnitt(int *z)
                                                {
                                                        int i, sum = 0;
                                                        float Durch;

                                                        for (i = 0;i < MAX;i++)
                                                                sum = sum + *(z+i);
                                                                Durch= (float)sum / MAX;

                                                        return Durch;
                                                }
```

## Maximum des Array´s ausgeben

```
// Programm MAXIMUM was den INDEX des Array´s ausgiebt der den größten Wert enthält

#include <stdio.h>

#define MAX 5



void lesen(int *);
int Maximum (int *);



        void main()
        {
                int Vorgang[MAX];

                lesen(Vorgang);

                printf("Maximum: %i\n", Maximum(Vorgang));
        }


                        void lesen(int *Zaehler)
                        {

                                int i;

                                for (i = 0; i < MAX ; i++)
                                {
                                printf ("Gebe Zahl ein");
                                scanf ("%i", Zaehler+i);
                                }
                        }


                                        int Maximum (int *z)
                                        {
                                                int i, temp;
                                                int maxi = -32000;

                                                for (i = 0;i < MAX;i++)
                                                {
                                                        printf("maxi %i, temp %i, i %i\n", maxi ,temp ,i);
                                                        printf("\n z+i %i", *(z+i));

                                                        if(* (z+i)> maxi)
                                                        {
                                                                temp = i;
                                                                maxi= *(z+i);
                                                        }

                                                }
                                                return (temp);
                                        }
```

## Durchschnitt berechnen

```
#include <stdio.h>

int summe(int*, int);
double Durchschnitt (int*, int);

void main()
{
	int Array[5];
	int i;

	for (i=0;i<5;i++)
	{
	printf("Wert eingeben: ");
	scanf("%i",Array+i);
	}
	printf ("Summe aller Werte: %i\n", summe(Array,i));
	printf ("Durchschnitt aller Werte: %f",Durchschnitt(Array,i));
}


int summe(int*p, int Anzahl)
{
	int i;
	int summe = 0;

	for (i= 0; i <Anzahl; i++)
	{
		summe = summe + *(p+i);
	}

	return (summe);
}

	double Durchschnitt (int*p, int Dimension)
	{
		int sum;

		sum = summe (p, Dimension);

		return ((double)sum / (double)Dimension);
	}
```

## Tauschen von 2 int Werten

```
#include <stdio.h>

void tauschen(int*,int*);

void main()
{
	int Zahl1 ;
	int Zahl2 ;

	printf("Zahl1 eingeben: ");
	scanf("%i",&Zahl1);

	printf("Zahl2 eingeben: ");
	scanf("%i",&Zahl2);

	//Ausgabe vor dem Tauschen
printf("Vorher %i %i\n",Zahl1,Zahl2);

	tauschen(&Zahl1,&Zahl2);
	//Ausgabe Nach dem Tauschen
printf("Nachher %i %i\n",Zahl1,Zahl2);
}

void tauschen (int*Z1, int*Z2)
	{
		int help;

		if (*Z1 > *Z2)
		{
			// hole mir den Wert der nach Z1 zeigt und übergebe diesen Wert in help usw.
			help = *Z1;
			*Z1= *Z2;
			*Z2 = help;
		}

	}
```

## Tauschen von 5 eingegebenen Werten

```
#include <stdio.h>

void tauschen (int*, int*);



void main()
{
                int Zahlen[5];
                int Zaehler;

                int i;
                int j;

                for (Zaehler = 0; Zaehler < 5; Zaehler++)
                {
                        printf("Zahl eingeben: ");
                        scanf("%i",Zahlen+Zaehler);

                        //Ausgabe vor dem Tauschen
                printf("Vorher %i \n",*(Zahlen+Zaehler));
                }

                                for (i=0; i<4; i++)
                                {
                                        for (j= i+1; j<5; j++)
                                        {
                                         tauschen(&Zahlen[i],&Zahlen[j]);
                                        }

                                }


                                for (i = 0;i<5;i++)
                                {
                                //Ausgabe Nach dem Tauschen
                                        printf("Nachher %i \n",*(Zahlen+i));
                                }

}



void tauschen (int*Z1, int*Z2)
        {
                int help;

                if (*Z1 > *Z2)
                {
                        // hole mir den Wert der nach Z1 zeigt und übergebe diesen Wert in help usw.
                        help = *Z1;
                        *Z1= *Z2;
                        *Z2 = help;
                }
```

```
}
```

## Strings erstellen und ausgeben:

```
#include <stdio.h>


void main()
{
	char szString[80];
	char szText[80];

	printf("Bitte String 1 eingeben :  ");
	//gets liest auch Leerzeichen ein! Aber hierbei kein %s schreiben
	gets(szString);


	printf("Bitte String 2 eingeben :  ");
	scanf("%s",szText);

	printf("\nDas war der 1 String  %s", szString);
	printf("\nDas war der 2 String  %s", szText);
}
```

## Befehle für Strings:

```
#include <stdio.h>

#include <string.h>


void main()
{

 char szTest[80];
 char szHelp[80];

 int i;

	//schreibt Hello in die szTest variable
	strcpy (szTest,"Hallo");
	strcpy (szHelp,"Mist");


	// Copiert szHelp in szTest
	strcpy (szTest, szHelp);

	//fügt beide Strings aneinander
	strcat (szTest,szHelp);

	//in die Variable i wird die länge der genutzten Zeichen übergeben
	i = strlen(szTest);

	//vergleicht Test mit Help
	strncmp(szTest,szHelp)
}
```

## Strings mit Funktionen:

```
#include <stdio.h>

void Einlesen(char*);
void Ausgeben(char*);

void main()
{
	char szText[80];

	Einlesen(szText);
	Ausgeben(szText);
}

	void Einlesen (char* pszWort)
	{
		printf("Bitte String eingeben");
		gets(pszWort);
	}

	void Ausgeben(char* pszWorte)
	{
		printf("%s",pszWorte);
	}
```

## String mit Funktionen und Pointern:

```
#include <stdio.h>

void Einlesen(char*);
void Ausgeben(char*);

void main()
{
	char szText[80];

	Einlesen(szText);
	Ausgeben(szText);
}


	void Einlesen (char* pszWort)
	{
		printf("Bitte String eingeben");
		gets(pszWort);
	}

	void Ausgeben(char* pszWorte)
	{
		char * pszPoi;

			pszPoi = pszWorte;

			for (pszPoi=pszWorte;*pszPoi;pszPoi++)
			{

				printf("%c",*pszPoi);
```

```
                        }
        }
```

## String rückwärts ausgeben lassen:

```
#include <stdio.h>

#include <string.h>

void Einlesen(char*);
void Ausgeben(char*);

void main()
{
        char szText[80];

        Einlesen(szText);
        Ausgeben(szText);
}


        void Einlesen (char* pszWort)
        {
                printf("Bitte String eingeben");
                gets(pszWort);
        }

void Ausgeben(char* pszWorte)
{

char* psz;
int len;
int i;

len = strlen(pszWorte);     //Erkennt wie lang der String ist
psz = pszWorte+(len-1);  // Setzt den Pointer auf die letzte stelle des INDEXès


for (i=len;i>0;i--)  // Schleife zum runterzählen des Indexes
{
        printf("%c",*psz); // Setzt den Buchstaben
        psz--;                  // zieht pro Durchlauf eine Stelle ab um zum nächsten Buchstaben zu gelangen

}
}
```

## Tauschen mit funktion  ostrupper keine vorhandenen Funktionen nutzen

```
#include <stdio.h>

void ostrcpy(char*, char*);

void main()
{
	char ziel[80];
	char quelle[80];

	printf("String eingeben :");
	scanf ("%s",quelle);

	printf("%s",quelle);

	//beachten das ziel und quelle wie in der funktion aufgereit werden
	ostrcpy(ziel,quelle);

	printf("%s",quelle);
	printf("%s",ziel);
}




// beachten das ziel und quelle im main funktionsaufruf auch so geschrieben werden
void ostrcpy(char *ziel, char *quelle)
{
	int i=0;
		while(*(quelle+i))
		{
			*(ziel+i)=*(quelle+i);
		i++;

		}
		*(ziel+i)='\0';
}
```

**Präprozessorderecktieven:**

Gekennzeichnet durch Raute #

z.B.
# include <stdio.h>

#include “test.h”  =>

NAME / Inhalt
#define MAX 20  => dies sind Konstante Variablen die überall nutzbar sind. Beipiel: in der Variablen deklaration int Wert [MAX]

**Beispiel:**

#define Printf  printf  =>  hier wird falls man das printf mal groß schreibt in ein kleines printf gewandelt

oder Berechnungen:

#define Quadrat(x)  (x*x)

#undef  => beendet die ausgewählte

**In Dateien schreiben:**

Dateizeiger anlegen

FILE *fp;

Fp = fopen („test.dat“,

oder mit Pfad angabe aber in Strings mit doppel Backslasch
fp = („c:\\temp\\test.txt“

**Moduse: für TEXTDATEIEN**

“r” =>öffnen zu lesen

“w“ => erstellt eine Datei zum schreiben „Achtung bestehende Dateien werden ohne Nachfrage
überschrieben“

“a“ => erstellt oder öffnet Datei zum schreiben „Überschreibt nicht schon vorhandene Dateien“ man kann
also auch in der Datei weiter schreiben

“w+“ => erstellt Datei zum lesen und schreiben

“a+“ => erstellt oder öffnet Datei zum lesen und schreiben Dateizeiger am Ende der Datei

**Beispiel: in Datei speichern**

```
#include <stdio.h>

void main()
{
	FILE *Text;
	Text= fopen("TestDatei.mk", "w");

	if (!Text)
	{
		printf("Datei kann nicht geöffnet werden");
	}
		else
		{
			fputs("Hallo",Text);
			fputs("\n",Text);
			fclose(Text);
		}
```

```
}
```

## Auslesen von Dateien:

```
#include <stdio.h>


void main()
{
	char uebergabeTXT[80];

	FILE *Text;
	Text= fopen("TestDatei.mk", "r");

	if (!Text)
	{
		printf("Datei kann nicht geöffnet werden");
	}
		else
		{
			fgets(uebergabeTXT,80,Text);
		do
		{
			fgets(uebergabeTXT,80,Text);
			printf("\n%s\n",uebergabeTXT);
		}while(!feof(Text));

		fclose(Text);
		}


}
```

## Einschreiben und Auslesen aus Datei in Funktionen:

```
#include <stdio.h>
#include <string.h>

void Einschreiben(char*);
void Auslesen(char*);

void main()
{
	char String[80];

	Einschreiben(String);

	Auslesen(String);
}

void Auslesen(char *klo)
{
	char uebergabeTXT[80];

	FILE *TxT;
	TxT = fopen("TestDatei.mk", "r");

	if (!TxT)
	{
		printf("Datei kann nicht geöffnet werden");
	}
		else
		{
			fgets(uebergabeTXT,80,TxT);
		do
		{	printf("\n%s\n",uebergabeTXT);
			fgets(uebergabeTXT,80,TxT);

		}while(!feof(TxT));

		fclose(TxT);
		}
}

void Einschreiben(char *klo)
{
    char eingabevar[80];

	FILE *Tx;
	Tx= fopen("TestDatei.mk", "a");

	if (!Tx)
	{
		printf("Datei kann nicht geöffnet werden");
	}
		else
		{
			printf("Bitte Text :");
			gets( eingabevar);
			fputs(eingabevar,Tx);
			fputs("\n",Tx);
			fclose(Tx);
```

```
                }

}
```

## Datei rein schreiben mit Menü:

```
#include <stdio.h>

#include <string.h>

void Einschreiben(char*);
void Auslesen(char*);


void main()
{
        char String[80];
        char Auswahl[2];

        do
        {
                printf("Was wollen sie tun ?\n  1- Datei lesen \n  2- Datei schreiben\n  0- Beenden");
                gets(Auswahl);

        if(*Auswahl=='2')
        {
                Einschreiben(String);
        }

        if(*Auswahl=='1')
        {
                Auslesen(String);
        }

        }while (*Auswahl!='0');
}


void Auslesen(char *klo)
{
        char uebergabeTXT[80];

        FILE *TxT;
        TxT = fopen("TestDatei.mk", "r");

        if (!TxT)
        {
                printf("Datei kann nicht geöffnet werden");
        }
                else
                {
                                fgets(uebergabeTXT,80,TxT);
                        do
                        {       printf("\n%s\n",uebergabeTXT);
                                fgets(uebergabeTXT,80,TxT);

                        }while(!feof(TxT));

                        fclose(TxT);
                }
}
```

```
void Einschreiben(char *klo)
{

   char eingabevar[80];

        FILE *Tx;
        Tx= fopen("TestDatei.mk", "a");



        if (!Tx)
        {
                printf("Datei kann nicht geöffnet werden");
        }
                else
                {
                        printf("Bitte Text :");
                        gets( eingabevar);
                        fputs(eingabevar,Tx);
                        fputs("\n",Tx);
                        fclose(Tx);
                }


}
```

## Schreiben und lesen mit Menü Version 2

```
#include <stdio.h>
#include <string.h>

#define LOESCHEN "2w"
#define SCHREIBEN "3a"

void Einschreiben(char*);
void Auslesen(char*);
void Einschreiben2(char*);

void main()
{
	char String[80];
	char Auswahl[3];

	do
	{
		printf("Was wollen sie tun ?\n  1- Datei lesen \n  2w- vorhandene Datei löschen und neu
			anlegen\n  3a- in vorhandener Datei weiter schreiben\n  0- Beenden\n:");
		gets(Auswahl);


	if(!strcmp(Auswahl,LOESCHEN))
	{
		Einschreiben2(String);
	}

	if(!strcmp(Auswahl,SCHREIBEN))
	{
		Einschreiben(String);
	}

	if(*Auswahl=='1')
	{
		Auslesen(String);
	}

	}while (*Auswahl!='0');
}


void Auslesen(char *klo)
{
	char uebergabeTXT[80];

	FILE *TxT;
	TxT = fopen("TestDatei.mk", "r");

	if (!TxT)
	{
		printf("Datei kann nicht geöffnet werden");
	}
		else
		{
				fgets(uebergabeTXT,80,TxT);
```

```
                        do
                        {       printf("\n%s\n",uebergabeTXT);
                                fgets(uebergabeTXT,80,TxT);

                        }while(!feof(TxT));

                        fclose(TxT);
                }
}


void Einschreiben(char *klo)
{

   char eingabevar[80];

        FILE *Tx;
        Tx= fopen("TestDatei.mk", "a");



        if (!Tx)
        {
                printf("Datei kann nicht geöffnet werden");
        }
                else
                {
                        printf("Bitte Text :");
                        gets( eingabevar);
                        fputs(eingabevar,Tx);
                        fputs("\n",Tx);
                        fclose(Tx);
                }



}


void Einschreiben2(char *klo)
{

   char eingabevar[80];

        FILE *Tx;
        Tx= fopen("TestDatei.mk", "w");



        if (!Tx)
        {
                printf("Datei kann nicht geöffnet werden");
        }
                else
                {
                        printf("Bitte Text :");
                        gets( eingabevar);
                        fputs(eingabevar,Tx);
                        fputs("\n",Tx);
                        fclose(Tx);
                }
```

}

## Binärdateien erstellen:

### Moduse: für Binärdateien

“rb” =>öffnen zu lesen

“wb“ => erstellt eine Datei zum schreiben „Achtung bestehende Dateien werden ohne Nachfrage überschrieben“ (löscht)

“ab“ => erstellt oder öffnet Datei zum schreiben „Überschreibt nicht schon vorhandene Dateien“ man kann
also auch in der Datei weiter schreiben

“wb+“ => erstellt Datei zum lesen und schreiben (löscht)

“ab+“ => erstellt oder öffnet Datei zum lesen und schreiben Dateizeiger am Ende der Datei

### Beispiel:

```
#include <stdio.h>

void main()
{

FILE *f;
char szs[] = "Hallo";
int anzahl;
int i=5;
f= fopen("c:\\test.txt","wb");
anzahl = fwrite(szs,sizeof(szs),1,f);   // sizeof ermittelt die größe der Variablen oder einen Datentyp ermittelt es die größe des Datentyps
anzahl = fwrite(&i,sizeof(int),1,f);

fclose(f);
if (!f)
        {
                printf("Datei kann nicht geöffnet werden");
        }
        else
        {
        f= fopen("c:\\test.txt","rb");
        }
//anzahl= fwrite (&i,sizeof(int),1,f);   // 1 ist die Anzahl der zulesenden Zeichen und f  steht für die Datei
anzahl= fread(&i,sizeof(int),1,f);
fclose(f);
}
```

## Dateizeiger setzen

```
fops_t  a;
a=100;    // 100 ist die Anzahl der Bytes
fsetpos(f,a);
a= fgetpos(f, &a);
```

**oder  einfacher**

```
fseek(f,0,SEEK_SET);          // 0 Byte ist die Distance vom Dateianfang(SEEK_SET) von wo
                              // SEEK_END vom Ende und SEEK_CUR aktueller stand des Dateizeigers
                              // rewind(f); setzt den Dateizeiger auf den Datei Anfang
```

**löschen von Dateien**

```
remove(„C:\\test.txt“);
```

**Umbenennen**

```
rename(“test.txt”,”schnulli.dat”);
```

**Fehler suchen**

```
a= ferror(f)       // f  steht für das Dateihändel welches am Anfang angelegt wurde
```

**typedef** ist ähnlich dem struct  man kann variablen oder andere sachen ändern.

```
typedef  int  integer  // hier wurde aus dem in C bekannten int   integer gemacht

typedef char  schnulli  // nun kann man anstatt char  schnulli verwenden
```

## Suchstring in Datei vorhanden und wo Ausgabe?

```
#include <stdio.h>
#include <string.h>

void anlegen(char*);

void main()
{
	char String[80];

	anlegen(String);
}


void anlegen(char *wc)
{

	char suche[3];
	unsigned int i;
	char Menue[3];

	FILE *Datei;
	char Text[80];

	do
	{
	printf(" Datei anlegen: 1\n Suchen in Datei : 2 \n Beenden : 0\nWas soll es denn sein: ");
	gets(Menue);

	if (*Menue == '1')
	{
	printf("Bitte Text eingeben : ");
	gets(Text);

	Datei = fopen("c:\\Hallo.txt","w");
	fputs(Text,Datei);
	fclose(Datei);
	}

	if (*Menue == '2')
	{

		Datei = fopen("c:\\Hallo.txt","r");


		if (!Datei)
		{
			printf("Datei konte nicht geöffnet werden");
		}
		else
		{
			printf("Wonach wollen Sie suchen?");
```

```
                        gets(suche);


                fgets(Text,80,Datei);

                for(i=0;i<strlen(Text);i++)
                if(!strncmp(Text+i,suche,strlen(suche)))
                        break;


                }
        }
                printf("%i",i);

                fclose (Datei);

        }while(*Menue !='0');


}
```

## Auslesen und schreiben von Binärdateien

```
#include <stdio.h>
#include <string.h>

void main()
{
        char szHelp[20];
        char szs[]="Hallo Welt";
FILE *fp;
fp = fopen("Test.dat","wb");
fwrite(szs,sizeof(char),strlen(szs),fp);
fclose(fp);


fp = fopen("Test.dat","rb");
fseek (fp,6,SEEK_SET);                          //oder fseek(fp,-7,SEEK_END);
fread(szHelp,sizeof(char),4,fp);
szHelp[4]='\0';
printf("Gelesen: %s",szHelp);
fclose(fp);
}
```

## Strukte

```
#include <stdio.h>
#include <string.h>


struct Personaldaten
{
	char nachname[40];
	char vorname[20];
	int	persnr;
	double gehalt;
};


void main()
{
	struct Personaldaten person;		// = {"Hurtig",Hugo",1,2079}; so erpart man sicht das unten
	strcpy(person.nachname,"Hurtig");
	strcpy(person.vorname,"Hugo");
	person.persnr = 1;					// bei Zahlentypen kein strcpy
	person.gehalt= 2095.17;


	// Bildschirmausgabe
	printf(" Vorname: %s\n Name:    %s\n Persnr:  %i\n Gehalt: % .2f\n",
person.vorname,person.nachname,person.persnr,person.gehalt);
}
```

**Strukte mit Zeigern**

```
#include <stdio.h>
#include <string.h>


struct Personaldaten
{
	char nachname[40];
	char vorname[20];
	int	persnr;
	double gehalt;
};


void main()
{
	struct Personaldaten person;  // = {"Hurtig",Hugo",1,2079}; so erpart man sicht das
unten

	struct Personaldaten *poi;
	poi = &person;

	strcpy(person.nachname,"Hurtig");
	strcpy(person.vorname,"Hugo");
	person.persnr = 1;                              // bei Zahlentypen kein strcpy
	person.gehalt= 2095.17;




	// Bildschirmausgabe
	printf(" Vorname: %s\n Name:    %s\n Persnr:  %i\n Gehalt: % .2f\n", poi-
>vorname,poi->nachname,poi->persnr,poi->gehalt);
}
```

Speicher allocieren mit calloc

```
#include <stdio.h>


int *pi;
float *pf
        //Speicherplatz 100,größe von int
pi= (int*) calloc (100, sizeof(int));
pf=(float*)calloc(5,sizeof(float));

*pi =30;  // in den ersten integer des angelegten pointers
pi[1]=5;  // so könnte man es auch schreiben es wird hier aber die 2 Adress des Pointers mit
              dem Wert 5 belegt
*(pi+2)=7;// so könnte man es auch schreiben hier wird in die 3 Adresse der Wert 7
               übergeben
```

# C++ Programmierung

## Scpe Operator **:: variablen Name** die lokal ist

```
#include <stdio.h>

int x=5;

void main()
{
        int x = 20;

        printf("%d  %d",x,::x);

}
```

## Funktionenen

```
#include <stdio.h>

void funktion (int a= 5, int b= 10, int c= 15);


void main()
{
        funktion(7,9); // hir werden für a und b andere Werte übergeben
}

void funktion (int a, int b, int c)
{
        printf("a= %i  b= %i  c= %i ",a,b,c);
}
```

```
#include <stdio.h>
```

## Referenzvariablen

```
void main()
{
        int a = 20;
        int &b= a;

        b=50;

        printf("A ist %i",a);   // A ist 50
}
```

## Referenzen und Funktionen

```
#include <stdio.h>


void tausche(int&, int&);


void main()
{
        int x = 5, y=3;

        tausche(x,y);


}



void tausche (int &a, int &b)
{
        int help;


        help = a;
        a=b;
        b=help;
}
```

This-> ist ein Zeiger der auf das Objekt zeigt Strukturen und Klassen besitzen dieses

This-> Variablenname

## Konstanten anlegen

```
Const int c = 5
```

// ersetzt das define da dies zu gefährlich war, spätere Änderungen der Werte der Constanten sind nicht möglich

## Inline Funktionen

```
Inline int quadrat (int x)
{
        return (x * x);
}


a=5;
a= quadrat (a++);
```

a= 26

Variablen => sind in c++ Atribute

Funktionen => sind Methoden

Klasse => ist eine Sammlung von Atributen und Methoden

## Hallo Welt

```
# include <iostream.h>

void main()
{
        // Bildschirmausgabe
        cout << "Hallo C++!" << endl;
}
```

**cout und cin sind wie scanf und printf !!!**

**endline =>  macht eine Leerzeile und leert den Puffer**

**cout.flush();  => wie getchar**

**strings einlesen:**

**gets  =>  ist jetzt cin.getline (x,sizeof(x)-1);**

**cin.getline => für komplette Strings**

**cin.get (c);  => für 1 Zeichen**

## Addieren

```
#include <iostream.h>


int addiere (int a, int b);
int c;

void main()
{
	int i;
	int a[5];



	for (i=0;i<5;i++)
	{
	cout<< " Zahl eingeben\n";

	cin >>a[i];
	}


		for (i=0;i<5;i++)
		{
			cout << " Zahl  "<< a[i] << endl;

			cout << " Addition :" << addiere (a[i], a[i]) << endl;

		}

}


int addiere( int a, int b)
{
	c=a+b;

	return c;
}
```

## Klassen in C++

```
#include <iostream.h>



class Schwein
{
private:
        int Gewicht;
        int Saettigung;
       bool Geschlachtet;

public:
        Schwein (int,int);

        void zustand(void);
        void bewegen(int);
        void fressen (int);
        void rennen(int);
};




 Schwein ::Schwein (int a, int b)  // Konstruktor
{
        Gewicht = a;
        Saettigung = b;
        Geschlachtet = false;
}



void Schwein ::zustand(void)         // Funktion zustand der Class Schwein zuordnen
{
        cout << "Gewicht =" << Gewicht << endl;

        cout << " Saetigung =" << Saettigung << endl << endl;
}




void Schwein :: bewegen (int meter)
{
        if (Gewicht >= 100 || Geschlachtet)
        {
                cout <<"Urrrg Schwein wurde zu Wurst verarbeitet" << endl << endl;
                Geschlachtet = true;
                return;
        }
        else
        {
        Gewicht -= (meter/10)+1;
        Saettigung-= (meter/10)+10;
        }
```

```
	if (Saettigung > 50)
	{
	Gewicht -= (meter/10)+1;
	Saettigung-= (meter/10)+10;
	}
	else
	{
		cout << " Oink Oink ich hab Hunger" << endl <<endl;
	}
}



void Schwein :: fressen(int Kartoffel)
{
	if (!Geschlachtet)
	{
	Saettigung += ((Kartoffel/10)+1)*5;
	Gewicht += Kartoffel;
	}
}



void Schwein ::rennen (int Geschwindigkeit)
{
	if(Saettigung > 50 && !Geschlachtet)
	{
		Gewicht -= (Geschwindigkeit/10)+5;
		Saettigung-= (Geschwindigkeit/10)+20;
	}
	else
	{
		cout <<" Keine Lust mehr zu renne hab Hunger oder wurde geschlachtet"<< endl;
	}
}



void main()
{
	Schwein S(80, 100);				//Zuweisung Werte für S der Klasse Schwein

	S.zustand();				// Instanz

	for (int i =0;i< 10;i++)
	{
	S.bewegen (0);
	S.fressen (5);
	S.rennen (0);
	S.zustand ();
	}
}
```

## Klasse Tulpe:

```
#include <iostream.h>
#include <string.h>


class Tulpe
{
	private:
		bool bluete;
		char bluetenfarbe[30];
		int groessestengel;
		int dickestengel;

	public:
		Tulpe(int,int);				// Konstruktor anlegen

		void nahrungsaufnahme(int,int,int);
		void bluehen(void);
		void zustand(void);		// zum Ueberwachen
};




Tulpe :: Tulpe(int gs, int ds)			// erster Zugriff bei Programm start (Konstruktor)
{
	groessestengel = gs;
	dickestengel = ds;
	bluete = false;
	strcpy (bluetenfarbe,"");
}




void Tulpe :: nahrungsaufnahme(int wasser,int duenger,int sonne)
{
	if (groessestengel >= 400)
	{
		cout << "Blume ist hinueber" << endl;
		bluete = false;
		strcpy (bluetenfarbe,"Grau");
		return;
	}
		else if(groessestengel >= 300)
		{
			cout << "Knospe wird zur Bluete" << endl;
			groessestengel += (wasser + duenger + sonne)+10;
			dickestengel += ((wasser + duenger + sonne)/100)+1;
		}

			else if(groessestengel >= 200)
			{
				cout << "Knospe ist da !!! " << endl;
				bluete = true;
				groessestengel += (wasser + duenger + sonne)+10;
				dickestengel += ((wasser + duenger + sonne)/100)+1;
```

```
                        }
        else
        {
                groessestengel += (wasser + duenger + sonne)+10;
                dickestengel += ((wasser + duenger + sonne)/100)+1;
        }
}



void Tulpe :: zustand(void)          // Funktion zustand der Class Tulpe zuordnen
{
        cout << "Groesse der Tulpe in mm =" << groessestengel << endl;
        cout << "Dicke des Stengels in mm =" << dickestengel << endl;
        cout << "Bluete =" << bluete << endl;
        cout << "Blueten Farbe =" << bluetenfarbe << endl << endl;
}



void Tulpe :: bluehen (void)
{
        if (bluete == true)
        {
                strcpy (bluetenfarbe, "Gelb");       // Gelb wird der char Variablen in Class Tulpe zugewiesen
        }
}



void main()
{
        Tulpe B(1,1);                // Konstruktor bekommt startwerte

        B.zustand();

        for (int i=0;i<15;i++)
        {
        B.nahrungsaufnahme(10,5,12);
        B.bluehen ();
        B.zustand ();
        }
}
```

## Pointer

```
Int *p;

p = new(int)            // reserviert für den Zeiger p die Größe eines Integers

*p = 5                  // Zeiger wird der Wert 5 übergeben
```

## Classen mit Pointer

```
#include <iostream.h>

class xy
{
        private:
int a;
int b;
public:
        void show(void);
         xy (int ,int);          //Konstruktor
};

xy :: xy (int x, int y)   // Konstruktor
{
        a=x;
        b=y;
}

void xy :: show (void)
{

        cout << "A=" << a << " B = " << b << endl;
}


void main ()
{
        xy  x(2,7);                     // Konstruktor (Instanz der Classe)
        xy *px;                         // pointer wird mit der Class initialisiert
        x.show();               // funktion Show wird über x aufgerufen

px= new (xy)(5,9);    // für Zeiger px wird Speicherplatz der Größe der Klasse bereitgestellt
px -> show ();        // Teiger px ruft die Funktion show auf
}
```

## Dateien (Filehandling)

| | | |
|---|---|---|
| ios :: | in | öffnen zum lesen |
| | out | öffnen zum schreiben |
| | ate | öffnen zum anhängen |
| | app | öffnen und Dateiende suchen und da weiter schreiben |
| | trumc | Dateilänge auf 0 setzen |
| | nocreate | nur existierende öffnen |
| | noreplace | nur neu erstellen keine existierenden öffnen |

### Datei öffnen zum schreiben und lesen über Menü

```
#include <iostream.h>
#include <fstream.h>
#include <string.h>

void main ()
{
        ofstream raus;                          // schreibt in Datei Out File Stream
        ifstream rein;                          // liest aus der Datei

        int Menue;

        cout << " Bitte 1 für schreiben oder 2 für lesen eingeben" << endl;
        cin >> Menue;

if (Menue==1)
{
        raus.open("Test.txt",ios::out);       // 1 Dateiname , (wie) hier zum schreiben

        if (!raus)
        {
                cout << " Fehler" << endl;
        }
        else
        {
                int stop = 0;                   // für die Schleife
                char szs[100];

                while(!stop)
                {
                        cin.getline (szs,100);       // Zeile einlesen

                        if ((strlen(szs)==1) && (szs[0] =='.')) // lese in die Zeile solange ein bis Punkt
                                                                eingegeben wird
                        {
                                stop=1;
                        }
                        else
                        {
                                raus << szs << "\n";         // schreibe nach Return in die nächste Zeile
                        }
                }
                raus.close();                           // schließe die Datei
```

```
        }

}


if (Menue == 2)
{
        rein.open("Test.txt",ios::in);          // (1 Dateiname , (wie)) hier zum lesen

        if (!rein)
        {
                cout << " Fehler" << endl;
        }
        else
        {
                char szs[100];

                rein >> szs;

                while(!rein.eof () )
                {
                        cout << szs << endl;
                        rein >> szs;
                }
                rein.close();                           // schließe die Datei
        }
}
}
```

## 2 Klassen lesen und schreiben in Datei mit Menü

```
#include <iostream.h>
#include <string.h>
#include <fstream.h>


class inDatei
{
private:
	char Dateiname2[30];	// Variable für Dateiname

	ofstream schreiben;		// Variable zum schreiben

		char szs[100];		// Variable von Methode Schreiben

public:
	inDatei(char*);		//Konstruktor
	~inDatei(void);		//Destruktor

	void Schreiben(void)
	{
		int stop = 0;		// für die Schleife

		while(!stop)
		{
			cin.getline (szs,sizeof(szs));		// Zeile einlesen

			if ((strlen(szs)==1) && (szs[0] =='.')) // lese in die Zeile solange ein bis Punkt
								eingegeben wird
			{
				stop=1;
			}
			else
			{
				schreiben << szs << "\n"; // schreibe nach Return in die nächste Zeile
			}
		}
	}
};



class ausDatei
{
private:
	char Dateiname[30];		//Variable für Dateiname
	ifstream lesen;			// Variable zum lesen
	char szs[100];			// Variable für Methode lesen

public:
	ausDatei(char*);	// Konstruktor
	~ausDatei(void);	// Destruktor

	void Lesen(void)
	{
		lesen.clear();				//löscht die Dateiflags
		lesen.seekg(0,ios::beg);	//seekg setzt den Dateizeiger auf NULL
		lesen >> szs;				//zum lesen
```

```
                        while(!lesen.eof () )          // prüfe auf eof
                        {
                                cout << szs << endl; //Ausgabe von szs (Inhalt der Datei)
                                lesen >> szs;       // aus der Datei "Text.txt" Inhalt einlesen in szs
                }
        }

};


inDatei :: inDatei(char*y) //Konstruktor**********SCHREIBEN******************
{
        strcpy(Dateiname2,y);                   // Kopiert y siehe im main("Test.txt) in Variable Dateiname
        schreiben.open(Dateiname2,ios::out);  //öffnet "Test´.txt zum schreiben

        if(!schreiben)                          //Überprüfung ob Datei geöffnet wurde
        {
                cout <<"Fehler bei schreiben" << endl;
        }
}



ausDatei :: ausDatei(char *x)          //Konstruktor*************LESEN**************
{
        strcpy (Dateiname,x);                   // Kopiert x siehe im main("Test.txt) in Variable Dateiname
        lesen.open(Dateiname,ios::in);        //öffnet "Test´.txt zum schreiben

        if (!lesen)                                     //Überprüfung ob Datei geöffnet wurde
        {
                cout << "Fehler bei lesen" << endl;
        }
}




ausDatei :: ~ausDatei(void)                     //Destruktor---------(lesen)-----------
        {
                lesen.close();                          // schließe die Datei
        }


inDatei :: ~inDatei(void)               //Destruktor---------(schreiben)-------
        {
                schreiben.close();                      //schließen der Datei
        }



void main()
{
        int menue;                              //Variable für Menü

        do
        {
                cout << " 1- Lesen oder 2- Schreiben 3- Beenden" << endl;
                cin >> menue;                   //Einlesen der Eingabe in Variable Menü

                if (menue == 1)
                {
                ausDatei L("test.txt");
```

```
                L.Lesen ();
                }
                if (menue == 2)
                {
                inDatei S("test.txt");
                S.Schreiben ();
                }

        }while(menue!=3);          //Programm beenden wenn 3 eingegeben
}
```