1 Aufbau eines Programmes

```
#include <iostream> // Standart In-/ Output stream
#include <vector>   // Vector library
#include <cmath>    // Für math. funktionen
#include <time.h>   // Zeitmessung
#include "headerfile.h" // Einbiden Headerfile
#define N 10  // defines jeglicher art

//structs, functions, enums

int main(void)
{
//programm code
return 0;
}
```

2 Variablen

Group	Type names	Notes on size / precision
Character types	char	One byte 8 bits.
	char16_t	At least 16 bits
	char32_t	At least 32 bits.
	wchar_t	Largest character set
Integer types (signed)	signed char	Min 8 bits.
	signed short int	Min 16 bits.
	signed int	Min 16 bits.
	signed long int	Min 32 bits.
	signed long long int	Min 64 bits.
Integer types (unsigned)	unsigned char	Min 8 bits.
	ßhort int	Min 16 bits.
	ïnt	Min 16 bits.
	"long int	Min 32 bits.
	"long long int	Min 64 bits.

Mögliche Initialsisations von vaiablen

```
int x;
int x = 1;
int x (1);
int x {1};
```

2.1 Pointer und Referenzen als Rückgabewert und Parameterübergabe

Bei Variablenübergabe (call by value) werden Kopien übergeben, welche nicht verändert werden können.
Bei Referenzübergabe (call by reference) kann die Subroutine die Werte bleibend verändern.
Objekte einer Klasse und Strukturvariablen sollen immer by reference übergeben werden!

2.2 call by reference

```
void swap(int& a, int& b)
{
int tmp = a;
a = b;
b = tmp;
}
int main()
{
int x = 4;
int y = 3;
swap(x, y); // OK!
return 0;
}
```

```
void swap(int* a, int* b)
{
int tmp = *a;
*a = *b;
*b = tmp;
}
int main()
{
int x = 4;
int y = 3;
swap(&x, &y); // OK!
return 0;
}
```

2.3 call by value

```
void swap(int a, int b)
{
int tmp = a;
a = b;
b = tmp;
}
int main()
{
int x = 4;
int y = 3;
swap(x, y); // keine Ausw.
return 0;
}
```

bkljdfbkjsdfb asfoih sdafojps dfsdfjksdh fdfks dsdf sdf sdf sdfjoisjdf sd f
sdfj sodp f
df ihoüsdfjs df