

Datatypes

```
std::string
int
float
etc.
```

Arrays and List

```
int ar[size]
int arLength = sizeof(ar) / sizeof(
int);
```

Class Body File Paradigm // Compare Header

```
#include "Zaehlw erk .h"
Zaehlw erk ::Z aeh lwe -
rk(int ma){
zahl = 0;
max = ma;
}
void Zaehlw erk ::I -
nkr eme nti eren(){
if (zahl < max)
zahl++;
}
std::s tring Zaehlw -
erk ::A nze igen(){
return "der Zaehler ist
auf der " +st d:: to_ -
s tring (zahl);
}
```

Extensions / CPP File

```
#include
"Ringwerkzaehler.h"
Ringwe rkz aeh ler ::R -
ing wer kza ehler()
:Zaehl werk(){ }
```

Extensions / CPP File (cont)

```
> Ringwerkzaehler::Ringwerkza-
ehler(int ma) : Zaehlw erk(){ }
void Ringwerkzaehler::Inkreme-
ntieren(){
if (zahl < max)
zahl++;
else
zahl = 0;
}
```

bubbleSort Algorithmus

```
void bubblesort(int
*array, int length){
int j;
for (j = 0; j < length
- i - 1; ++j) {
if (array[j] > array[j
+ 1]) {
int tmp = array[j];
array[j] = array[j +
1];
array[j + 1] = tmp;
}
}
```

Compare Classes

```
bool operator < (classobj) {
return (cVar < obj.cVar); } //ln
Constructor
```

Class header File Paradigm

```
class Zaehlw erk
{
protected:
int zahl;
int max;
public:
```

Class header File Paradigm (cont)

```
> Zaehlw erk();
Zaehlw erk(int max);
~Zaehlw erk();
virtual void Inkrementieren();
virtual void Dekrementieren();
virtual std::string Anzeigen();
virtual void Zuruecksetzen();
private:
};
```

Extensions / Header File

```
class Ringwerkzaehler :
Zaehlw erk
{
public:
Ringwe rkz aeh ler();
Ringwe rkz aeh ler(int
ma);
~Ringw erk zae hler();
void Inkrem ent ieren()
override;
void Dekrem ent ieren()
override;
std::s tring Anzeigen()
override;
void Zuruec kse tzen()
override;
};
```

Thread Operations

```
std::this_thread::sleep_for(std::-
chrono::seconds(1));
```

String functions

```
printf("%s\n", text.c_str());
std::string s1(*, 20);
s1.append(s2);
int equ = s1.compare(2, 4, s2);
s1.clear();
s2.erase(1, 3);
s1.find_last_of(s2);
s.insert(2, " einString ", 0, 3);
s.length();
s.replace(2, 5, "anoString", 2,2);
std::string s3=s1.substr(2,3);
```

Data processing / Read file / char

```
std::ifstream f;
f.open (fi leName,
std::i os_ bas e:: out);
int counter = 0;
char t;
while (f.get(t))
if (t == c)
counter++;
f.close();
```

Struct Example

```
struct kontakt{
std::s tring name;
int nummer;
std::s tring toString()
{
std::s tring temp = name
+ " ";
temp += std::t o_s tri -
ng( num mer);
return temp;
}
};
```

Data processing / Write file

```
std::ofstream f;
f.open (fileName,
std::ios_base::out);
std::string temp =
" ";
temp = "joe ";
temp += std::string(
120);
f << temp << std::endl;
f.close();
```

Debugger / cout shorter

```
//Debug Messenger
#define DEBUG 1
#if DEBUG
#define LOG(x) std::cout
<< x << std::endl
#else
#define LOG(x)
#endif
system ("PA USE "); //
Just here to remember
```

Data processing / Read file / getLine

```
std::vector<kontakt>
kontaktListe(readFile(filename));
std::ifstream f;
f.open (fileName, std::ios_
base::out);
std::string t="";
int i = 0;
while (getline(f, t))
{
for each (char var in t)
{
if (var == ' ')
{
//Operation
}
else
i++;
}
i = 0;
}
f.close();
```

