

if-Anweisung

```

if (Bedingung) {
  Anweisung;
} else {
  Anweisung;
}

```

switch-case-Anweisung

```

switch (Ausdruck) {
  case Konstante1: Anweisung1; break;
  case Konstante2: Anweisung2; break;
  default: default-Anweisung; break;
}

```

while-Anweisung

```

while (Bedingung) {
  Anweisung;
}

```

do-while-Anweisung

```

do {
  Anweisung;
} while (Bedingung);

```

for-Anweisung

```

for (Ausdruck1; Ausdruck2; Ausdruck3) {
  Anweisung;
}

```

Pointer

```

datentyp *Zeigervariable;

```

Array

```

datentyp Name [Elementzahl];
datentyp Name [index 1][index 2]...;

```

## Datenstrukturen

```
struct Name {  
    datentyp_1 Name_1;  
    datentyp_n Name_n;  
};  
datentyp Funktion (struct Name *Name_n);
```

```
struct Name x[AnzahlElemente];
```

```
struct Name *Varname;
```

```
Varname = B x [0];
```

```
x[Element].Name_n;
```

## Klasse

```
class Name {
```

```
private:
```

```
    datentyp_1 Name_1(...);
```

```
public:
```

```
    datentyp_2 Name_2(...);
```

```
};
```

## Konstruktor

```
datentyp Name::Name_2(...) { return (...); }
```

```
Instanz Name Instanz;
```

```
Instanz Name_2(...);
```

```
Instanz Variable;
```