1 Inline code

1.1 Exercise

```
int main() {
  char msg[10];
  char *p;
  char msg2[]= "Hello";

  msg = "Bonjour";
  p   = "Bonjour";

  msg = p;
  p = msg;
}
```

2 lstinputlisting

2.1 Full

```
#include <cstddef>

#include <iostream>
#include <string>

int main() {
    static constexpr size_t kColumnCount = 4;
    std::cout << "ASCII-Table" << std::endl;

    for (size_t i = 32; i < 128; ++i) {
        std::cout.width(3);
        std::cout.fill('0');
        std::cout << i << " = 0x";
        std::cout.setf(std::ios::hex, std::ios::basefield);
        std::cout.setf(std::ios::uppercase);
        std::cout << i << ": ";
        std::cout.unsetf(std::ios::hex);
        std::cout << static_cast<char>(i) << '\t';

        if (i % kColumnCount == kColumnCount - 1) {
            std::cout << std::endl;
        }
    }
```

```
    std::cout << std::endl;
    return 0;
}
```

2.2 firstline & lastline

```
int main() {
    static constexpr size_t kColumnCount = 4;
    std::cout << "ASCII-Table" << std::endl;
```

2.3 linerange

```
int main() {
    static constexpr size_t kColumnCount = 4;
    std::cout << "ASCII-Table" << std::endl;
```

3 Images

3.1 Image drawio

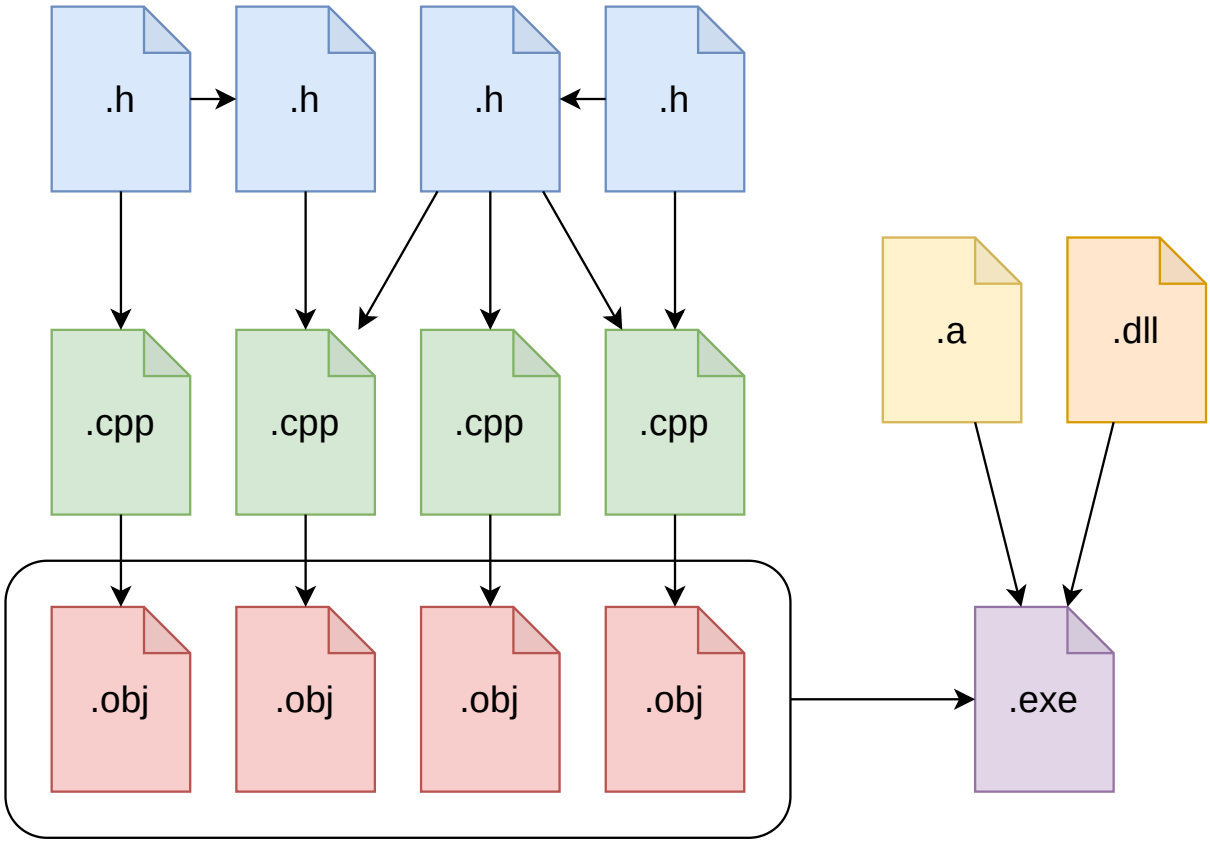

Figure 1: Image description

3.2 Image direct

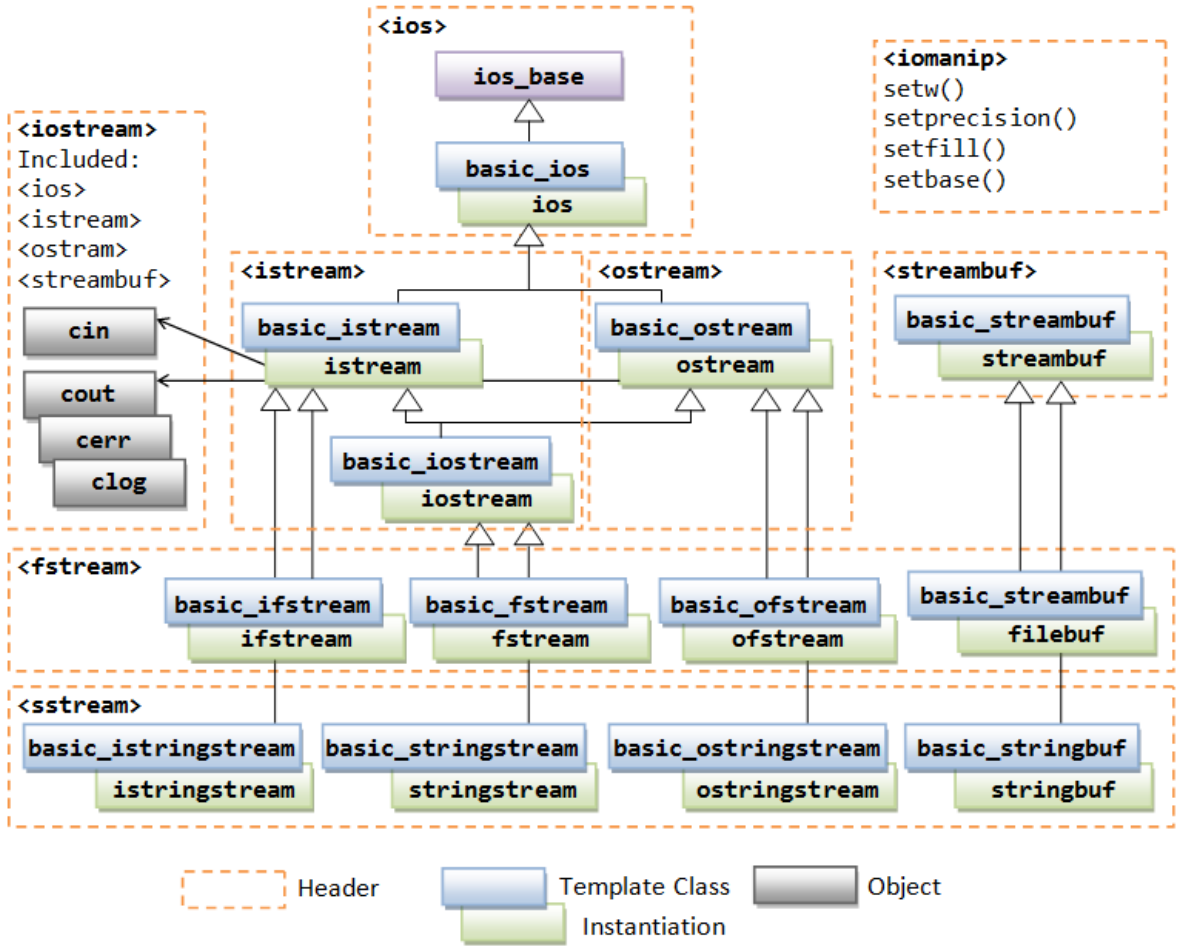

Figure 2: Image description

3.3 Image tikz

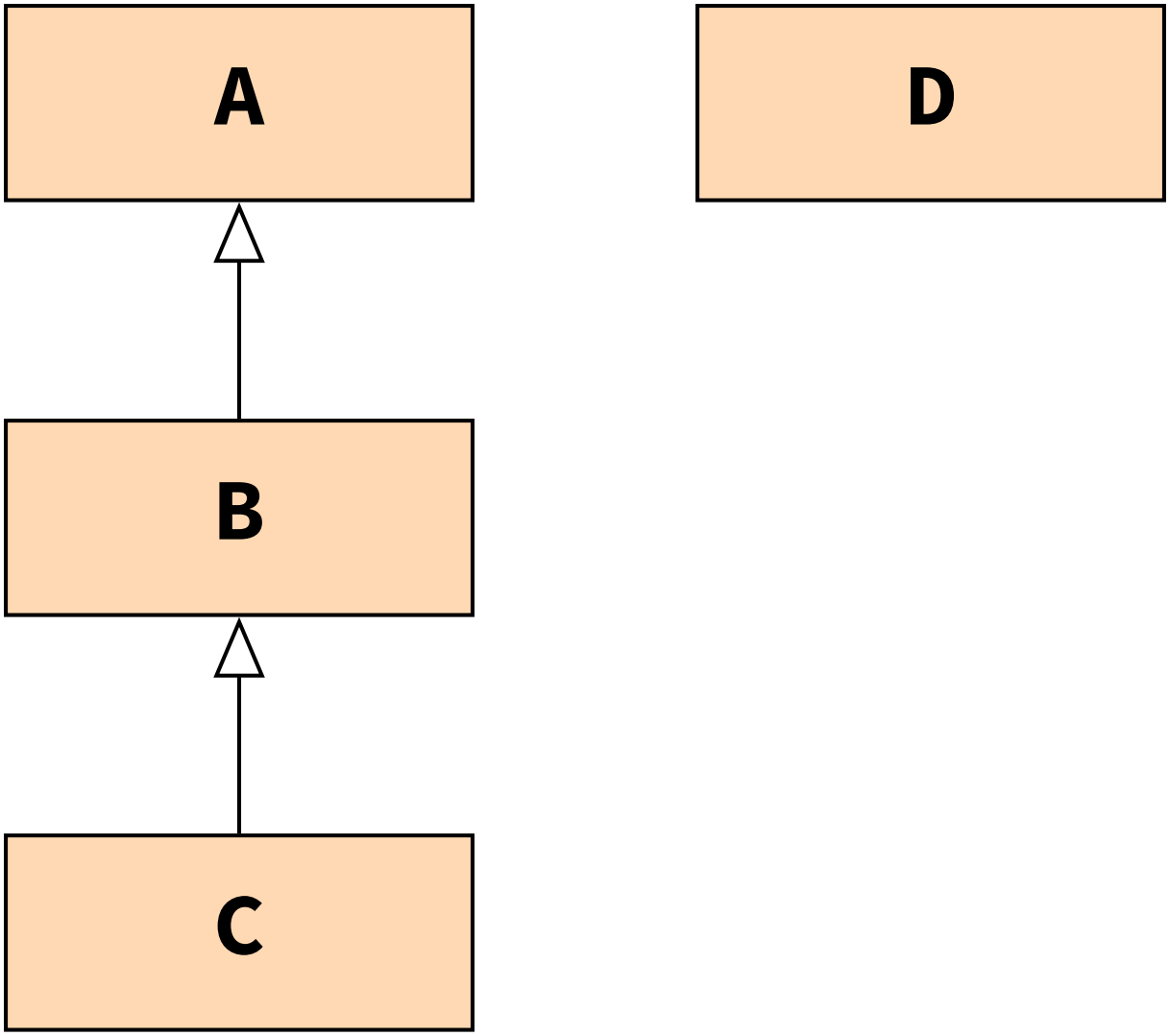

Figure 3: Image description

4 Formulas

Die Multiplikation zweier Matrizen A und B, $R = A \times B$ ist wie folgt definiert: $r_{i,j} = \sum_{k=1}^{v} a_{i,k} b_{k,j}$, wobei A eine $u \times v$, B eine $v \times w$ und R eine $u \times w$ Matrix sind.

Zum Beispiel: $$\begin{pmatrix} a_{0,0} & a_{0,1} & a_{0,2} \\ a_{1,0} & a_{1,1} & a_{1,2} \end{pmatrix} \times \begin{pmatrix} b_{0,0} \\ b_{1,0} \\ b_{2,0} \end{pmatrix} = \begin{pmatrix} a_{0,0} \times b_{0,0} + a_{0,1} \times b_{1,0} + a_{0,2} \times b_{2,0} \\ a_{1,0} \times b_{0,0} + a_{1,1} \times b_{1,0} + a_{1,2} \times b_{2,0} \end{pmatrix} = \begin{pmatrix} r_{0,0} \\ r_{1,0} \end{pmatrix}$$