

Repetitorium Test Teil 2

```
#include <iostream>
using namespace std;

class X {
    int x;
    int y;
public:
    X(int a=8, int b=7) : x{a+1}, y{b-1} {}
    void print() {
        cout << x << y;
    }
    X change() {
        return X {x-1, y+1};
    }
};

int main()
{
    X x{6};
    x.change().print();
    return 0;
}
```



Repetitorium Test Teil 2

```
#include <iostream>
using namespace std;

int f(int n) {
    cout << n;
    return (n < 10) ? f(n+1) : n*2;
}

int main()
{
    int n {8};
    cout << f(n);

    return 0;
}
```



Repetitorium Test Teil 2

```
#include <iostream>
#include <string>
using namespace std;

char f(string s) {
    s += 'X';
    return s.at(1);
}

int main()
{
    string s {"REGE"};
    cout << f(s);
    for (const auto& c : s)
        cout << c;

    return 0;
}
```



Repetitorium Test Teil 2

```
#include <iostream>
#include <vector>
using namespace std;

int main()
{
    vector<int> v {87,61,35,48,65};

    for (int i {4}; i>2; i=i-1) {
        int k {4};
        do {
            v.at(k) = v.at(k-1);
            k = k - 1;
        } while (k>0);
    }

    for (int i{0}; i<5; i=i+1)
        cout << v.at(i);

    return 0;
}
```



Repetitorium Test Teil 2

```
#include <iostream>
using namespace std;

int f(int& k, int n) {
    cout << k << n;
    k = k - 1;
    n = n + 1;
    cout << k << n;
    return k + n;
}

int main()
{
    int k {2}, n {1};

    cout << k << n;
    cout << f(k, n);
    cout << k << n;
    return 0;
}
```



Repetitorium Test Teil 2

```
#include <iostream>
using namespace std;

class X {
public:
    int x;
    X(int x=2) : x{x} {}
    void print() {cout << x;}
    X operator+(int rop) const {
        return rop + x;
    }
    int operator+(const X& rop) const {
        return x+rop.x;
    }
};

X operator+(int lop, const X& x) {
    return 2*lop + x.x;
}

int main()
{
    X x;
    cout << (1+x) + (x+4);
    x.print();
    return 0;
}
```



Repetitorium Test Teil 2

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string s {"REGE"};

    int i{0};
    for (const auto& c : s)
        switch (c) {
            case 'G': ++i;
            case 'R': cout << c;
                    break;
            default: cout << s.at(i);
        }

    cout << i;
    return 0;
}
```



Repetitorium Test Teil 2

```
#include <iostream>
#include <vector>
using namespace std;

void f(vector<int>& v, int i) {
    v.push_back(v.at(i)%10);
    if (i) f(v, i-1);
    v.push_back(v.at(i)/10);
}

int main()
{
    vector<int> v {87, 61, 35};
    size_t i{v.size()-1};

    f(v, i);

    for (const auto& i : v)
        cout << i;

    return 0;
}
```



Repetitorium Test Teil 2

```
#include <iostream>
using namespace std;

class X {
    int a;
    int b;
public:
    X(int a=8, int b=7) : a{a}, b{b} {}
    void print() {
        cout << a << b;
    }
    void add() {
        a += b--;
    }
};

int main()
{
    X x;
    x.add();
    x.print();
    return 0;
}
```



Repetitorium Test Teil 2

```
#include <iostream>
#include <string>
using namespace std;

string f(string &s) {
    string temp{s + "A"};
    s += temp;
    return temp;
}

int main()
{
    string s{"BC"};

    cout << f(s);
    cout << s;

    return 0;
}
```



Repetitorium Test Teil 2

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string s {"REGE"};
    string t {"STILL"};

    for (size_t i{0}, j{t.size()-1}; i<4; ++i, --j) {
        t[j] = s[i];
    }

    cout << t;

    return 0;
}
```



Repetitorium Test Teil 2

```
#include <iostream>
using namespace std;

class X {
    int x;
public:
    X(int x=8): x{x} {}
    void print() {
        cout << x;
    }
    int change(X y) {
        ++x, --y.x ;
        return x + y.x;
    }
};

int main()
{
    X x, y{7};
    cout << x.change(y);
    x.print();
    y.print();
    return 0;
}
```



Repetitorium Test Teil 2

```
#include <iostream>
using namespace std;

class X {
public:
    int x;
    X(int x=8) : x{x} { }
    void print() {cout << x;}
    int operator+(const X& rop) const {
        return x+rop.x;
    }
    X operator+() {
        return X{2*x};
    }
};

X operator+(int n, const X& x) {
    return X{n*n + x.x};
}

int main()
{
    X x;
    cout << x + (+x);
    x.print();
    return 0;
}
```



Repetitorium Test Teil 2

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string s {"REGE"};

    cout << s.at(1);
    cout << s;
    s.at(3) = s.at(2);
    cout << s.at(2);
    cout << s;

    return 0;
}
```



Repetitorium Test Teil 2

```
#include <iostream>
#include <vector>
#include <string>
using namespace std;

int main()
{
    vector<string> v {"REGE", "STILL", "RUHIG"};

    cout << v.at(2).at(2);

    string &s{v.at(1)};
    s.push_back(s.at(2));
    s.at(2) = 'X';

    cout << v.at(1);

    return 0;
}
```

