

```

// Einfacher Datentyp Bruch
// Destruktor und Copy-Konstruktor

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

class Bruch {
private:
    int zaehler;
    int nenner;
    int kennzeichen;
    static int Nummer;
public:
    Bruch (int, int);
    // Destruktor
    ~Bruch () {
        cout << "Bruch No. " << kennzeichen <<
            " zerstört!" << endl;
    }

    // Zuweisungsoperator
    Bruch& operator= (const Bruch&);

    // Copy Konstruktor
    Bruch (const Bruch&);

    void print ();

    Bruch operator * (Bruch);
    Bruch operator *= (Bruch);
}; // Bruch

```

```

//Initialisierung
int Bruch::Nummer = 0;

Bruch::Bruch (int z, int n) {
    // Nur sinnvolle Nutzung
    zaehler = z;
    nenner = n;
    kennzeichen = ++Nummer;
    cout << "Bruch No. " << kennzeichen <<
        " geschaffen!" << endl;
}

Bruch::Bruch (const Bruch& rs) {
    zaehler = rs.zaehler;
    nenner = rs.nenner;
    kennzeichen = ++Nummer;
    cout << "Bruch No. " << kennzeichen <<
        " in Copy-Konstruktor benannt!" << endl;
}

Bruch& Bruch::operator= (const Bruch& rs) {
    if (this != &rs) {
        zaehler = rs.zaehler;
        nenner = rs.nenner;
    }
    cout << "Zuweisung ausgeführt!" << endl;
    return *this;
}

void Bruch::print () {
    cout << zaehler << '/' << nenner << endl;
} // print

```

```
Bruch Bruch::operator * (Bruch b) {  
    return Bruch (zaehler * b.zaehler, nenner * b.nenner);  
}/*
```

```
Bruch Bruch::operator *= (Bruch b) {  
    this->zaehler *= b.zaehler;  
    this->nenner *= b.nenner;  
return *this;  
}/*=
```

```
int main() {  
    Bruch x (2, 3);  
    cout << "  x = "; x.print ();  
  
    Bruch a (7, 3);  
    cout << "  a = "; a.print ();  
  
    Bruch y (x);  
    cout << "  y = "; y.print ();  
  
    Bruch ya = a;  
    cout << "  ya = "; ya.print ();  
  
    Bruch z = Bruch (5, 7) * Bruch (10, 7);  
  
    cout <<"  z = "; z.print ();  
  
    z = Bruch (3, 2) * Bruch (4, 3);  
    cout << "  z = "; z.print ();  
return 0;  
}/*main
```

```
/* Ausgabe
```

```
Bruch No. 1 geschaffen!  
x = 2/3  
Bruch No. 2 geschaffen!  
a = 7/3  
Bruch No. 3 in Copy-Konstruktor benannt!  
y = 2/3  
Bruch No. 4 in Copy-Konstruktor benannt!  
ya = 7/3  
Bruch No. 5 geschaffen!  
Bruch No. 6 geschaffen!  
Bruch No. 7 geschaffen!  
Bruch No. 5 zerstört!  
Bruch No. 6 zerstört!  
z = 50/49  
Bruch No. 8 geschaffen!  
Bruch No. 9 geschaffen!  
Bruch No. 10 geschaffen!  
Bruch No. 8 zerstört!  
Zuweisung ausgeführt!  
Bruch No. 10 zerstört!  
Bruch No. 9 zerstört!  
z = 12/6  
Bruch No. 7 zerstört!  
Bruch No. 4 zerstört!  
Bruch No. 3 zerstört!  
Bruch No. 2 zerstört!  
Bruch No. 1 zerstört!  
  
*/
```