

# C++: Big 3

---

Carsten Gips (HSBI)

Unless otherwise noted, this work is licensed under CC BY-SA 4.0.

# Big Three

```
class Dummy {  
public:  
    Dummy(int a=0);  
    Dummy(const Dummy &d);  
    ~Dummy();  
    Dummy &operator=(const Dummy &d);  
private:  
    int value;  
};
```

```
Dummy::Dummy(int a): value(a) {}  
Dummy::Dummy(const Dummy &d): value(d.value) {}  
Dummy::~~Dummy() {}  
Dummy::Dummy &operator=(const Dummy &d) {  
    if (this != &d) { value = d.value; }  
    return *this;  
}
```

## delete this?

```
class Foo {  
public:  
    ~Foo() {  
        delete this;  
    }  
};
```

## C++11: *default* und *delete*

```
class Dummy {  
public:  
    Dummy() = default;  
    Dummy(int a) { value = a; }  
    Dummy(const Dummy &a) = delete;  
  
private:  
    int value;  
    Dummy &operator=(const Dummy &d);  
};
```

# Statische Methoden und Attribute

```
class Studi {  
    static int getCount();  
    static int count;  
};
```

```
int Studi::count = 0;  
  
int Studi::getCount() {  
    return Studi::count;  
}
```

# Konstante Methoden und Kontexte

```
class Studi {  
    int getCredits() const;  
    int getCredits();  
};
```

```
int Studi::getCredits() const {  
    return credits;  
}  
  
int Studi::getCredits() {  
    return credits;  
}
```

- Klassen: Destruktoren, Copy-Konstruktor, Zuweisungsoperator
- Vorsicht mit Default-\*strukturen/-operatoren
- Statische Methoden und Attribute:
  - Deklaration als `static` **nicht** in Implementierung wiederholen
  - Statische Attribute: Initialisierung außerhalb der Klasse!
- Konstante Methoden und Kontexte
  - `const` gehört zur Signatur der Methode!
  - Konstante Methoden dürfen auf konstanten Objekten/Referenzen aufgerufen werden

# LICENSE



Unless otherwise noted, this work is licensed under CC BY-SA 4.0.