

## VII. Klassen und Objekte

### 3. Freunde

erkläre Funktion bzw. Klasse zum Freund:

`private` wird aufgehoben.

Wichtiges Beispiel: Überlade `>>` und `<<` und erkläre sie zum Freund einer Klasse

⇔ `"cout << v1;"` möglich

`class_ueber.cc`

Beachte:

```
cout << "Hallo";           ⇔           operator<< (cout,"Hallo");
cout << "Hallo" << " Karlsruhe";  ⇔  operator<< (operator<< (cout,"Hallo"), " Karlsruhe");
```

### 4. Konstruktoren und Destruktoren

`class_planet_2/3/4/5/6.cc`

- Idee: Initialisierung von Klasse bei Erzeugung; vermeide zufällige Initialisierung
- Default-Konstruktor: `<Name der Klasse>()`;  
⇔ automatischer Aufruf; nicht automatisch erzeugt
- Konstruktor kann überladen werden ( $\rightarrow$  Argumente)

## VIII. Zeiger/Pointer

### 1. Motivation und Notation

`pointer_1.cc, pointer_2.cc, pointer_swap.cc`

- Syntax: `<Typ> * <Name>;`
- `*`: Inhaltsoperator ⇔ `&`: Adressoperator