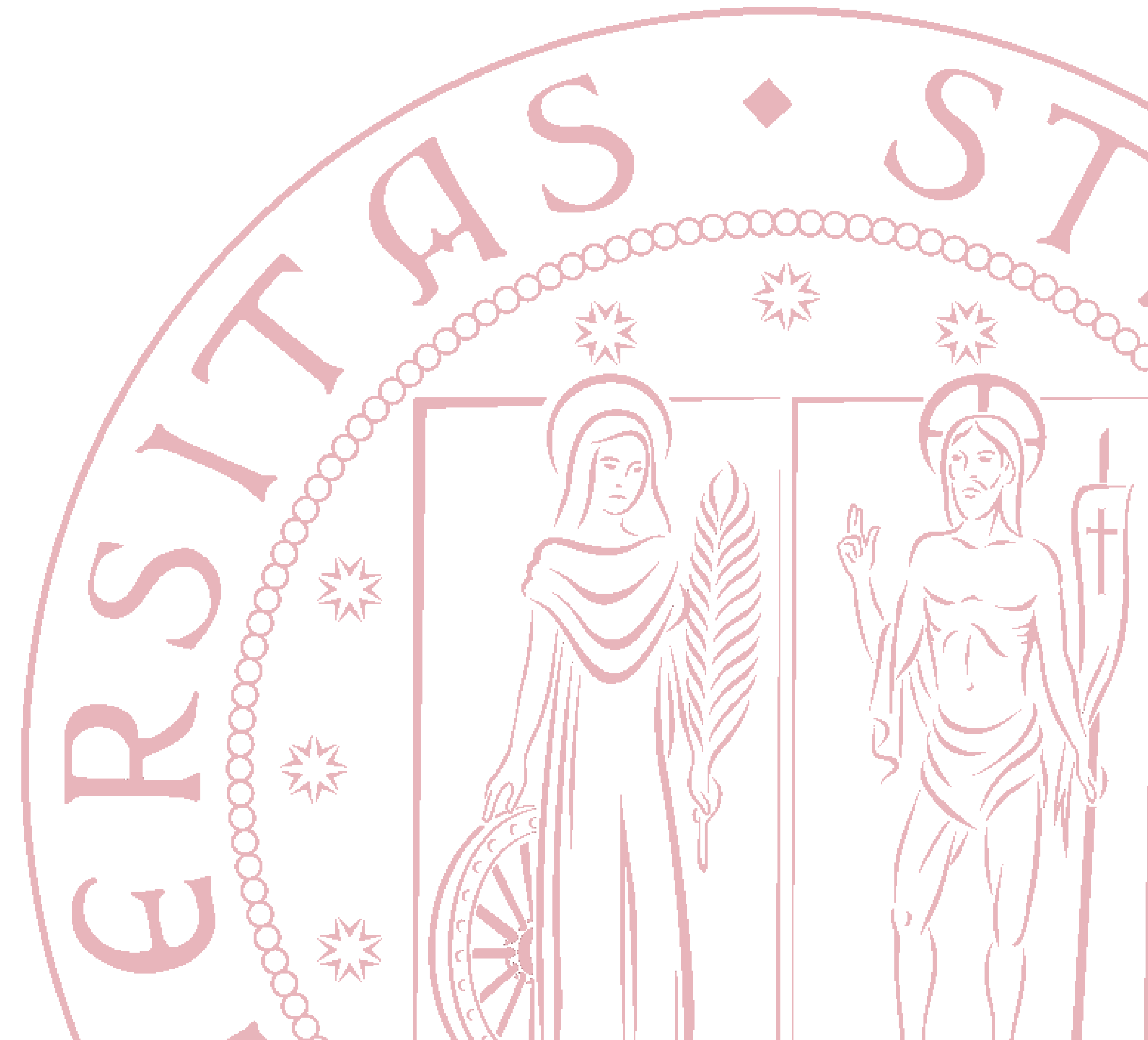


## 7.3 – Puntatore this

Libro di testo:

- Capitolo 17.10, 19.2.5



# Agenda

- Il puntatore `this`
- Caratteristiche e utilizzo



# Self-reference

- Abbiamo visto l'utilizzo di `this` nell'assegnamento di copia:

```
vector& vector::operator=(const vector& a)
{
    double* p = new double[a.sz];
    copy(a.elem, a.elem+a.sz, p);
    delete[] elem;
    elem = p;
    sz = a.sz;
    return *this;
}
```

**ritorna una self-reference  
dell'oggetto stesso**

# Puntatore **this**

- **this** è un puntatore all'oggetto stesso
- È un parametro implicito dell'oggetto, generato automaticamente
- Viene usato quando all'interno di una funzione membro:
  - È necessario accedere all'oggetto corrente
  - È necessario ritornare un riferimento all'oggetto corrente
- Ad esempio:
  - Assegnamento di copia
  - Assegnamento di spostamento
  - ...

# Puntatore this

- **this** è immutabile
- Non può in nessun caso essere modificato

```
struct S {  
    // ...  
    void mutate (s* p)  
    {  
        this = p;           // errore!  
        // ...  
    }  
};
```

# Utilizzo di `this`

Ritornare una reference all'oggetto corrente:

```
vector& vector::operator=(const vector& a)
{
    double* p = new double[a.sz];
    copy(a.elem, a.elem+a.sz, p);
    delete[] elem;
    elem = p;
    sz = a.sz;
    return *this;
}
```

Accedere all'oggetto corrente (da funzione membro):

```
class vector {
    int sz;
    double *elem;

    public:
        int size(void);
        // ...
};

int vector::size(void) {
    return this->sz;
}
```

- Non è strettamente necessario
- Questioni di stile