

## Ereditarietà pt 2:

Il principio di sostituibilità stabilisce che è possibile utilizzare un oggetto della sottoclasse dove è previsto un oggetto della superclasse

```
void f(B b){...}
```

```
int main() {
```

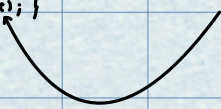
```
    B b;
```

```
    C c;
```

```
    f(b);
```

```
    f(c); }
```

c è della classe C, che è sottoclasse B



Aggiungere alla classe Persona il metodo

```
bool operator< (Persona p) {...}
```

```
class Persona {
```

```
protected:
```

```
    String name;
```

```
    String surname;
```

```
    String address;
```

```
public:
```

```
    Persona (String name, String surname) {
```

```
        this->name = name;
```

```
        this->surname = surname;
```

```
        this->address = "xxx";
```

```
    String getName ()
```

```
    {
```

```
        return name;
```

```
class Studente : public Persona {
```

```
private:
```

```
    int matricola;
```

```
    String email;
```

```
public:
```

```
    Studente (String nome, String cognome, int matricola) : Persona (nome, cognome) {
```

```
        this->matricola = matricola;
```

```
        this->email = nome + " " + cognome + "@univr.it";
```

```
    Studente () : Persona ("Marco", "Rossi")
```

```
    {
```

```
        this->matricola = 0;
```

```
        this->email = "xxx";
```

```
    }

    void stampa (ostream & dest) {
```

```
        Persona::stampa (dest);
```

però ad usare il costr con parametri





```

String getSurname() {

    return Surname;}

```

```

cout << "matricola: " << matricola << endl;

```

```

cout << "mail: " << mail; }

```

```

String getAddress(){

    return address;}

```

```

void setName (String name){

    this->name = name;}

```

```

void setSurname (String surname){

    this->surname= surname;}

```

```

void setAddress (String address ) {

    this-> address = address ;}

```

```

void stampa (ofstream & dest) {

    dest << name << " " << surname << endl;

    dest << indirizzo << endl; }

```

New!

```

bool operator< (Persona other) {

```

se fosse const

```

bool operator< (const Persona & other) const {

```

```

    if (cognome == other.cognome) {

```

```

        return nome < other.nome;}

```

```

    return cognome < other.cognome;}

```

```

int main() {

```

New!

```

    Studente s1 ("Vincenzo", "Arceri", 460366);

```

```

    Persona s2 ("Marco ", "Rossi ");

```

```

    Persona s3 ("Tizio ", "Verdi");

```

```

    cout << (s2 < s3) << endl;

```

```

    cout << (s2 < s1) << endl;

```

```

    =

```

```

    cout << (s1.operator<(s3)) << endl;

```