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 x) {...}

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 + "@.unipr.it"; }

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

                        this→matricola = 0;

                        this→email = "xxx"; }

                void stampa (ostream&  dest) {

                        Persona :: Stampa (dest);
```

forzo ad usare il costr con parametri

```
        string  getSurname () {

                return surname;}

        string  getAddress(){

                return address;}

        void  setName (string name){

                this → name = name;}

        void  setSurname (string surname){

                this →surname= surname;}

        void  set Address (string address){

                this→ address = address ;}

        void  stampa (ostream &  desr) {

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

                desr << indirizzo << endl; }

New!    bool  operator< (Persona other) {   se fosse const  →   bool operator< (const Persona &  other) const {

                if (cognome == other. cognome) {

                        return   nome < other. name;}

                return  cognome < other. cognome; }

    int main () {

new!    Studente  S1 ("Vincenzo", "Arceri", 460365);

    Persona  S2 ("Marco", "Rossi");

    Persona  S3 ("Tizio", "verdi");

    cout  <<  (S2 < S3) <<endl;

    cout  <<  (S2 < S1) <<endl;
        =
    cout << (S1.operator< (S2)) << endl;
```

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

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