

CLASSI: Point.h

```
class Point{
    float x;
    float y;
public:
    Point();
    Point(float, float);
    ~Point();
    void setx(float );
    void sety(float );
    float getx();
    float gety();
};
```

Membri privati

Costruttore (di default): ha lo stesso nome della classe e non può restituire un valore

Overloading di costruttore

Distruttore: ha lo stesso nome della classe preceduto da una tilde, non può restituire un valore né ricevere parametri

CLASSI: Point.cpp

```
#include <iostream>
#include "Point.h"
using namespace std;

Point::Point(){
    x=0; y=0;
    cout<<"Point()"<<endl;
}
Point::Point(float a, float b){
    x=a; y=b;
    cout<<"Point(float, float)"<<endl;
}
Point::~Point(){
    cout<<"~Point()"<<endl;
}
...
```

File header della classe

```
...
void Point::setx(float a){
    x=a;
}
void Point::sety(float b){
    y=b;
}
float Point::getx(){
    return x;
}

float Point::gety(){
    return y;
}
```

Scope resolution

CLASSI: test.cpp

```
#include <iostream>
#include "Point.h"
using namespace std;

int main(){
    Point p1;
    Point p2(3,4.1);

    //p1.x=6.3; //errore: compile-time

    cout<<"p1 "<<p1.getx()<<" " <<p1.gety()<<endl;

    p2.setx(-3.3);
    cout<<"p2 "<<p2.getx()<<" " <<p2.gety()<<endl;
}
```

Una possibile uscita

```
Point()
Point(float, float)
p1 0 0
p2 -3.3 4.1
~Point()
~Point()
```