Person　　　　　O

　　　　　　　　　　getters　setters

Person　　　　　　　　　　(real world)　　　　　　　　　(mapping abstraction)

Data Members

- [-] firstname : string
- [-] lastname : string

Member Functions

- [+] set_firstname(string fn) : void –> setter for firstname
- [+] get_firstname():string –> getter for firstname
- [+] set_lastname(string fn) : void –> setter for lastname
- [+] get_lastname():string –> getter for lastname
- [+] showInfo(): void –> print out its information

Constructor and Destructor

- Person() –> set default values of firstname and lastname as "unknown"
- Person(string fn, string ln) –> set firstname and lastname as fn and ln
- ~Person() –> print out that it has been deleted!

Source Code

```
#include <iostream>
using namespace std;

#ifndef _PERSON_
#define _PERSON_

class Person
```

Jun Wu

```
{
private:
  string firstname;
  string lastname;

public:
  Person();
  Person(string, string);
  void showInfo();
  void set_firstname(string fn);
  string get_firstname();
  void set_lastname(string ln);
  string get_lastname();
};
#endif
```

```
#include "person.h"

Person::Person()
{
  firstname="unknown";
  lastname="unknown";
}

Person::Person(string fn, string ln)
{
  firstname=fn;
  lastname=ln;
}

void Person::showInfo()
{
  cout << "Name: " << firstname << " " << lastname << endl;
}

void Person::set_firstname(string fn)
{
  firstname=fn;
}

string Person::get_firstname()
{
  return firstname;
}

void Person::set_lastname(string ln)
{
  lastname=ln;
```

```
}

string Person::get_lastname()
{
  return lastname;
}
```

```
#include <iostream>
using namespace std;

#include "person.h"

int main()
{
  Person *amy = new Person;
  amy->set_firstname("Amy");
  amy->set_lastname("Chang");

  cout << amy->get_firstname() << " " << amy->get_lastname() << endl;
  return 0;
}
```

```
all: person.o
    g++ main.cpp person.o -o main

person.o: person.cpp
    g++ -c person.cpp

clean:
    rm -f *.o main *.*~ *~
```