UML

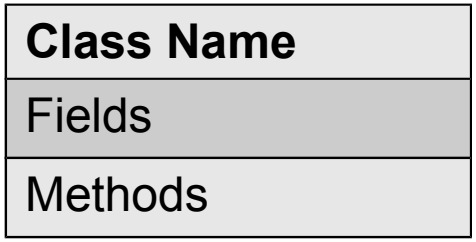
- UML is used as a graphical notation for describing programming concepts
- UML Stands for *Unified Modelling Languaee*
- It's not specific to Java
- For CSY2030 the main diagram is a CLASS DIAGRAM

Why use UML?

- Other developers can quickly look at the diagram and understand how the program works
  - How the application is structured
  - The classes, methods and fields which are available
  - How those classes and methods can be used
  - How classes are related

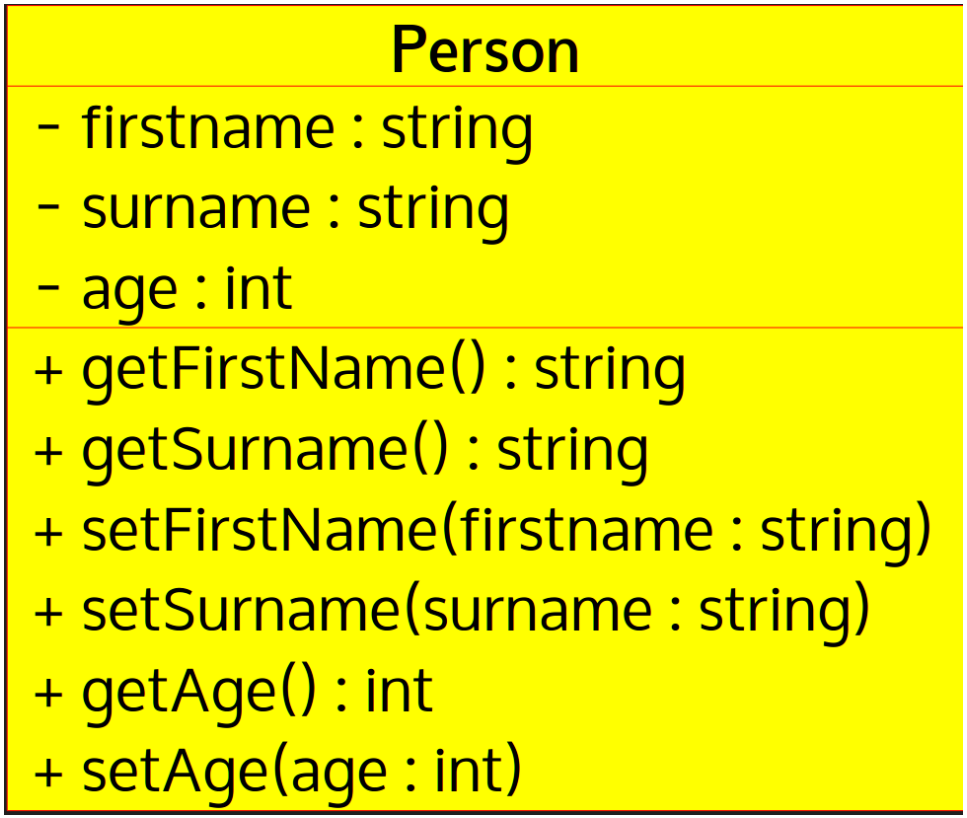
Class diagrams

- Class diagrams are boxes with three parts



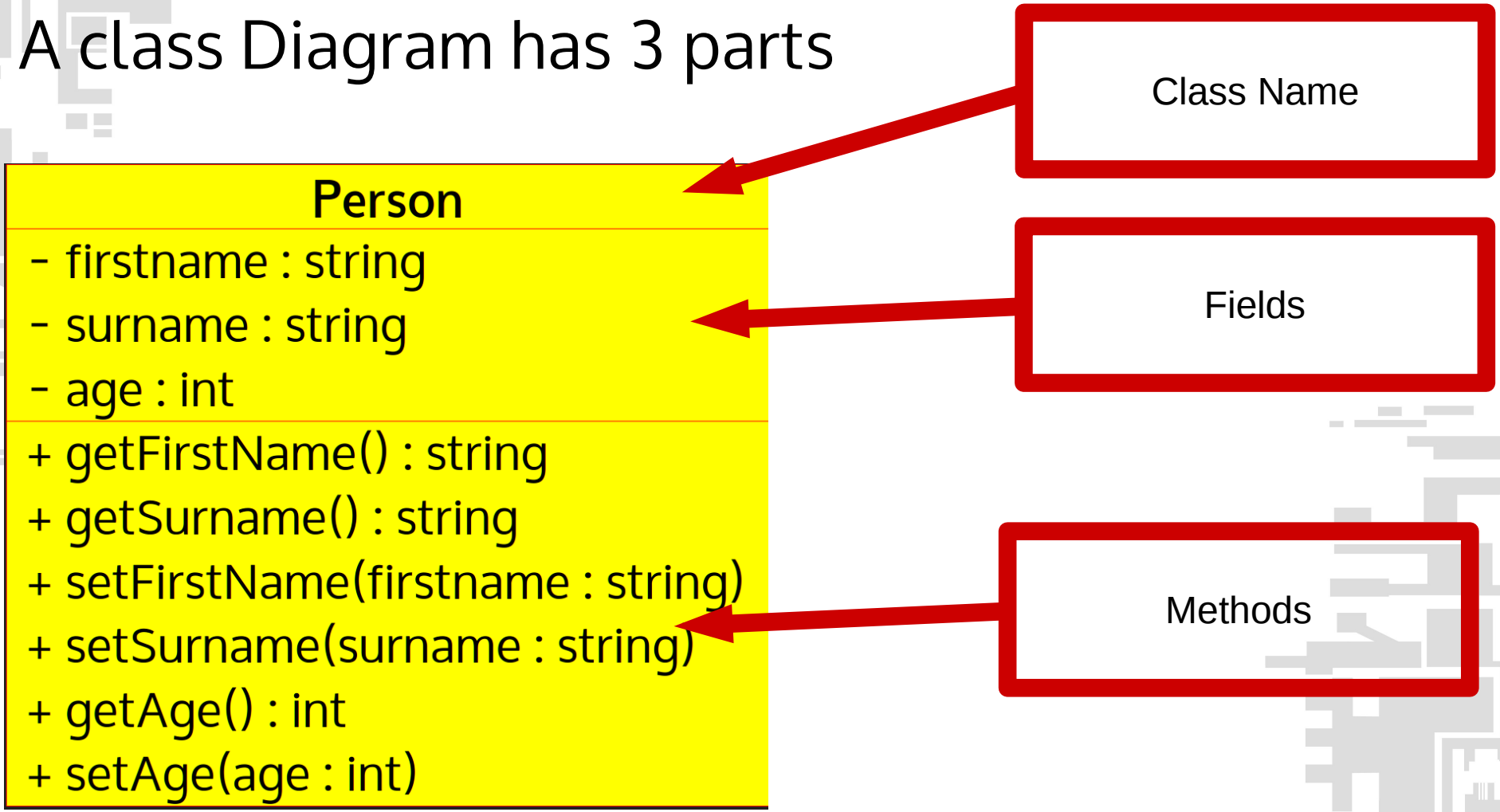
Class Diagrams

- UML contains a diagram format for describing classes



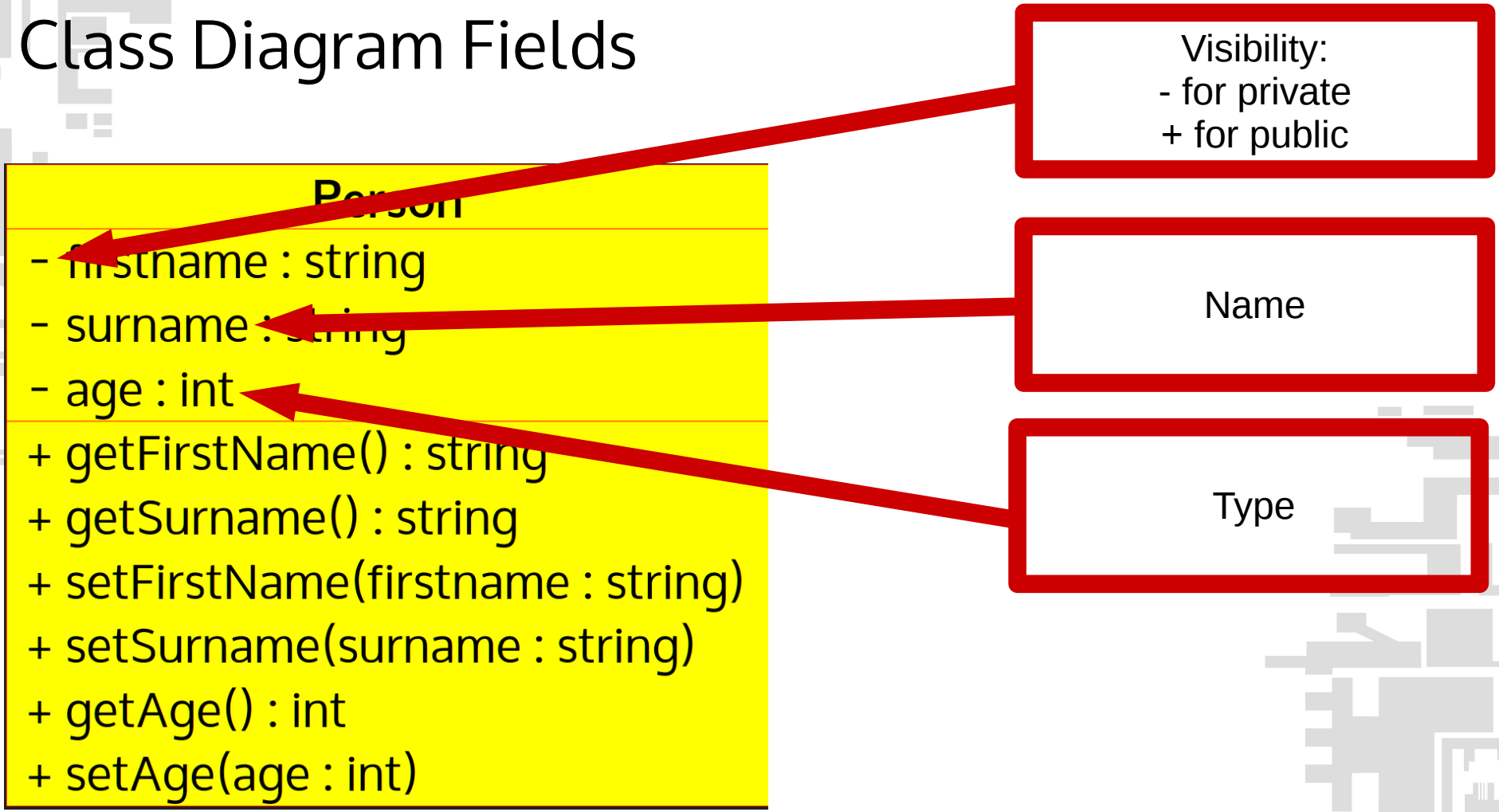
Class Diagram

- A class Diagram has 3 parts



Class Diagram

- Class Diagram Fields

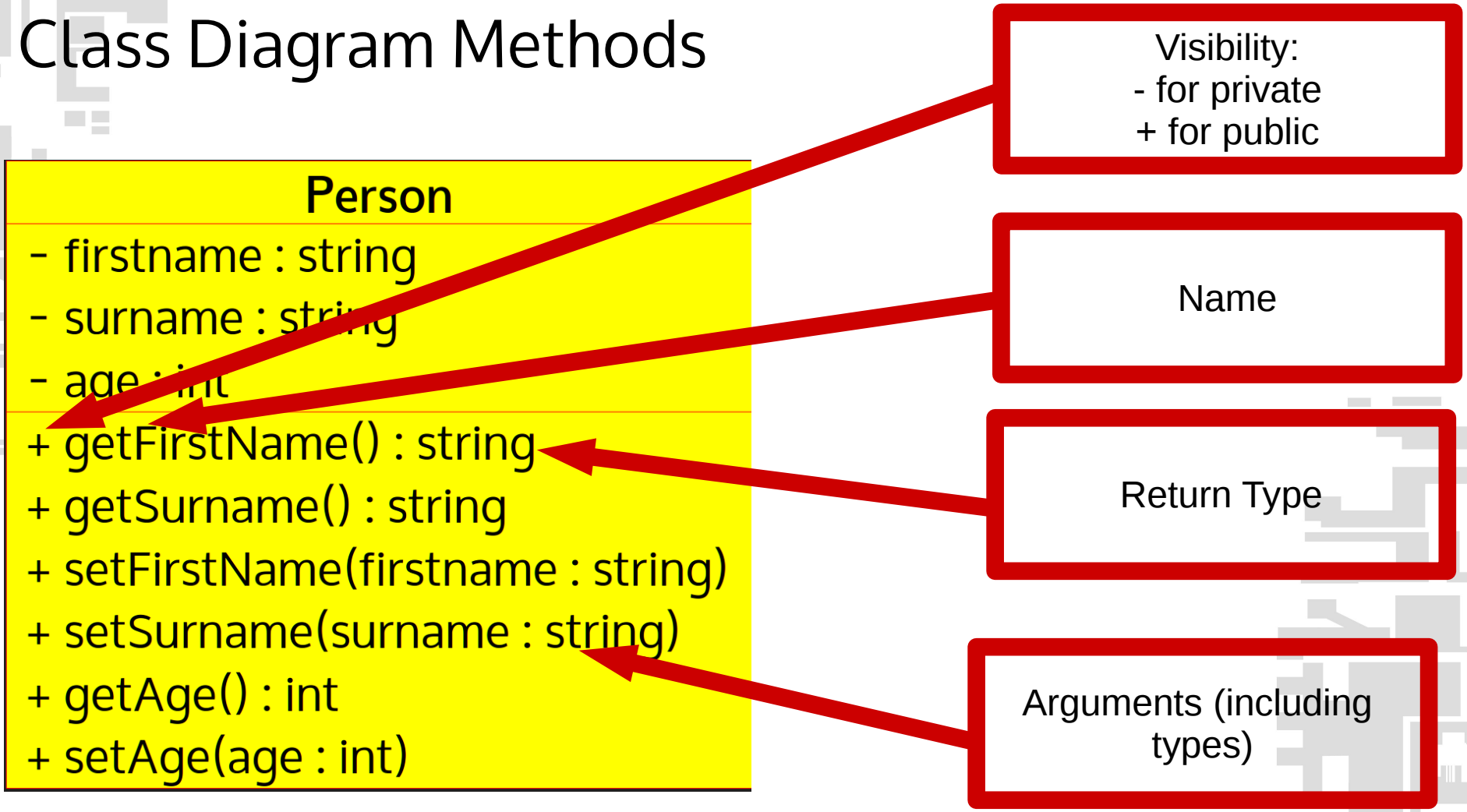


Class Diagrams

- The methods in class diagrams only show the *headers.* They do not show the logic.
- For each method the class diagrams show:
  - Name
  - Visibility
  - Arguments

Class Diagram

- Class Diagram Methods



Class API

- This is known as the *Class API*
- API stands for "Application Programmer Interface"
- Anyone using the class should only need an understanding of the API and not the *implementation* (how the method actually works)
- The programmer only needs to know the arguments and return type
- You will have seen this if you have ever looked up the documentation for a method in a programming language

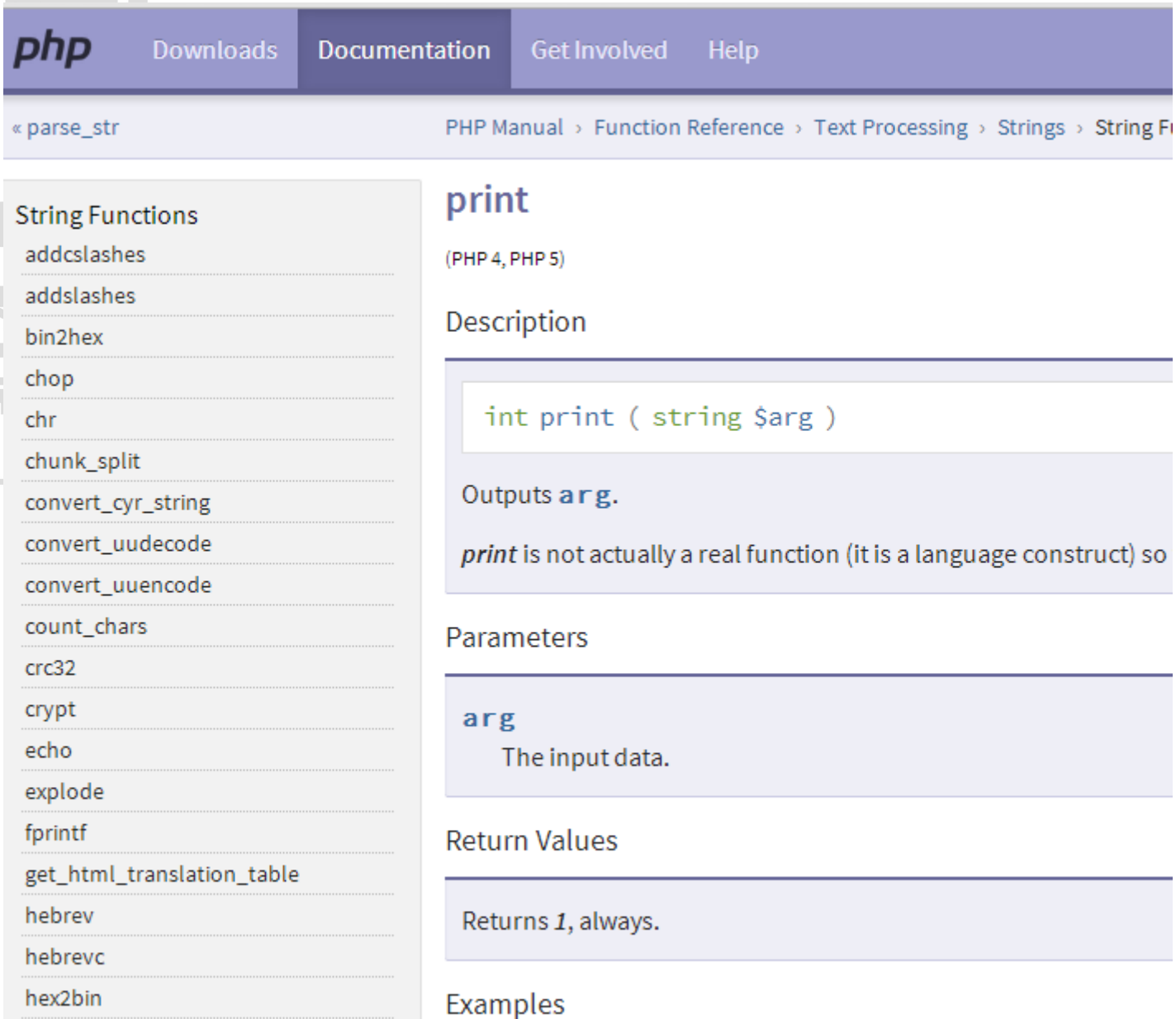
Class API

Method Summary	
static double	abs(double a) Returns the absolute value of a double value.
static float	abs(float a) Returns the absolute value of a float value.
static int	abs(int a) Returns the absolute value of an int value.
static long	abs(long a) Returns the absolute value of a long value.
static double	acos(double a) Returns the arc cosine of a value; the returned angle is in the range 0.0 through *pi*.
static double	asin(double a) Returns the arc sine of a value; the returned angle is in the range -*pi*/2 through *pi*/2.
static double	atan(double a) Returns the arc tangent of a value; the returned angle is in the range -*pi*/2 through *pi*/2.
static double	atan2(double y, double x) Returns the angle *theta* from the conversion of rectangular coordinates (x, y) to polar coordi
static double	cbrt(double a) Returns the cube root of a double value.
static double	ceil(double a) Returns the smallest (closest to negative infinity) double value that is greater than or equal to
static double	copySign(double magnitude, double sign) Returns the first floating-point argument with the sign of the second floating-point argument.
static float	copySign(float magnitude, float sign) Returns the first floating-point argument with the sign of the second floating-point argument.
static double	cos(double a)

Class API

Method Summary	
static double	abs(double a) Returns the absolute value of a double value.
static float	abs(float a) Returns the absolute value of a float value.
static int	abs(int a) Returns the absolute value of an int value.
static long	abs(long a) Returns the absolute value of a long value.
static double	acos(double a) Returns the arc cosine of a value; the returned angle is in the range 0.0 through *pi*.
static double	asin(double a) Returns the arc sine of a value; the returned angle is in the range -*pi*/2 through *pi*/2.
static double	atan(double a) Returns the arc tangent of a value; the returned angle is in the range -*pi*/2 through *pi*/2.
static double	atan2(double y, double x) Returns the angle *theta* from the conversion of rectangular coordinates (x, y) to polar coordi
static double	cbrt(double a) Returns the cube root of a double value.
static double	ceil(double a) Returns the smallest (closest to negative infinity) double value that is greater than or equal to
static double	copySign(double magnitude, double sign) Returns the first floating-point argument with the sign of the second floating-point argument.
static float	copySign(float magnitude, float sign) Returns the first floating-point argument with the sign of the second floating-point argument.
static double	cos(double a)

PHP print API



C# File Reading Method

File.ReadAllLines Method (String)

.NET Framework 4.5 | Other Versions | 0 out of 3 rated this helpful - Rate this topic

Opens a text file, reads all lines of the file, and then closes the file.

Namespace: System.IO
Assembly: mscorlib (in mscorlib.dll)

Syntax

C# | C++ | F# | VB

```
public static string[] ReadAllLines(
        string path
)
```

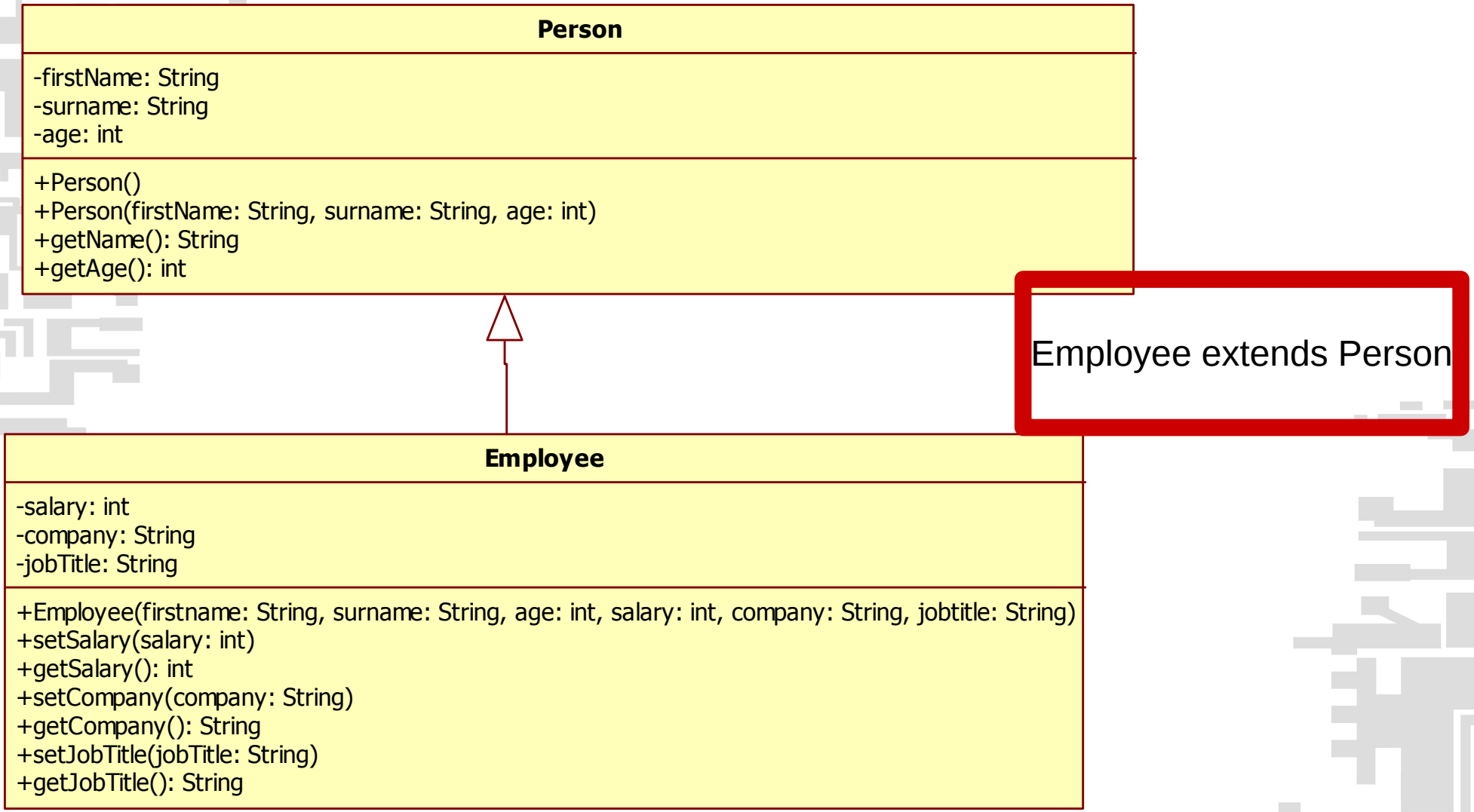
Parameters

path
Type: System.String
The file to open for reading.

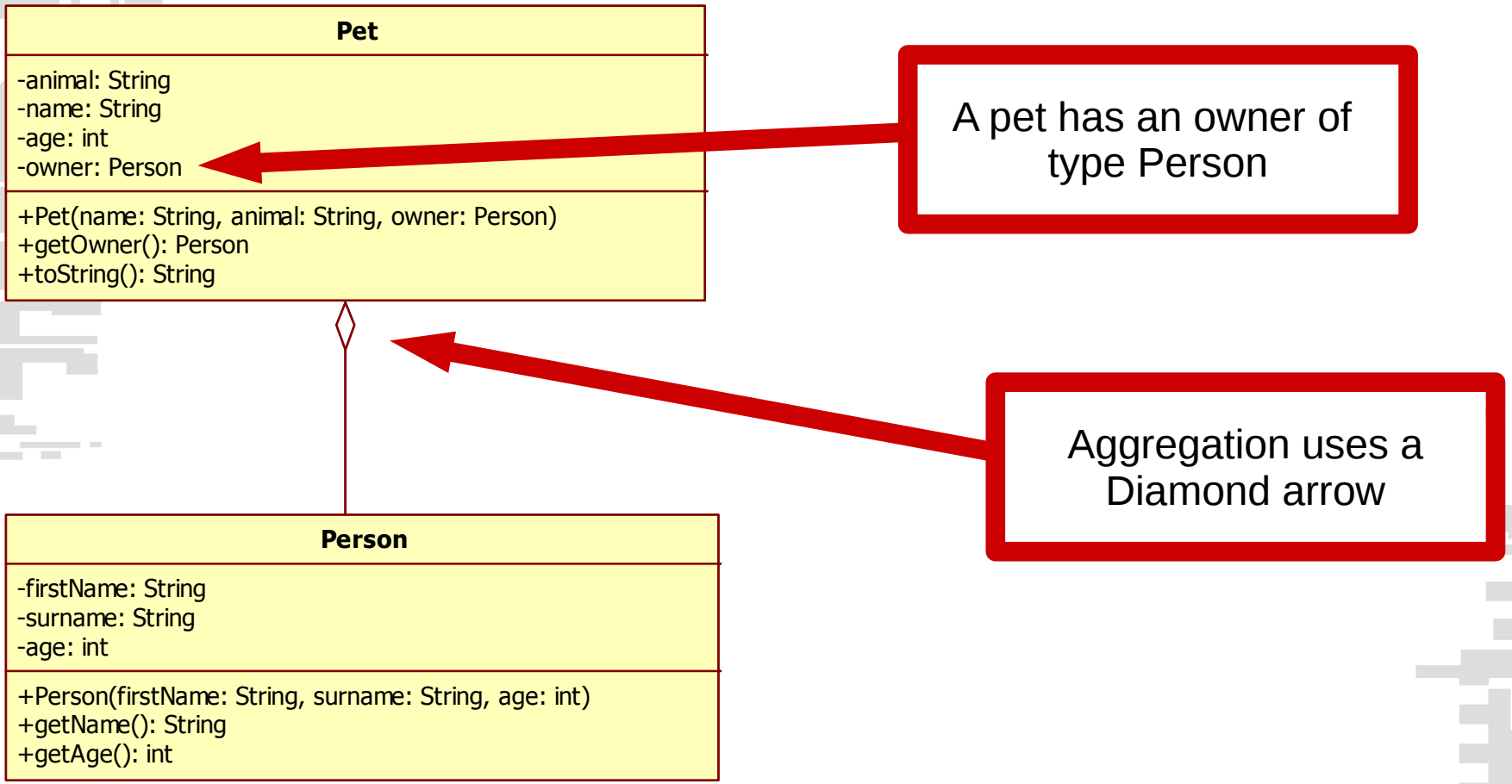
Class API

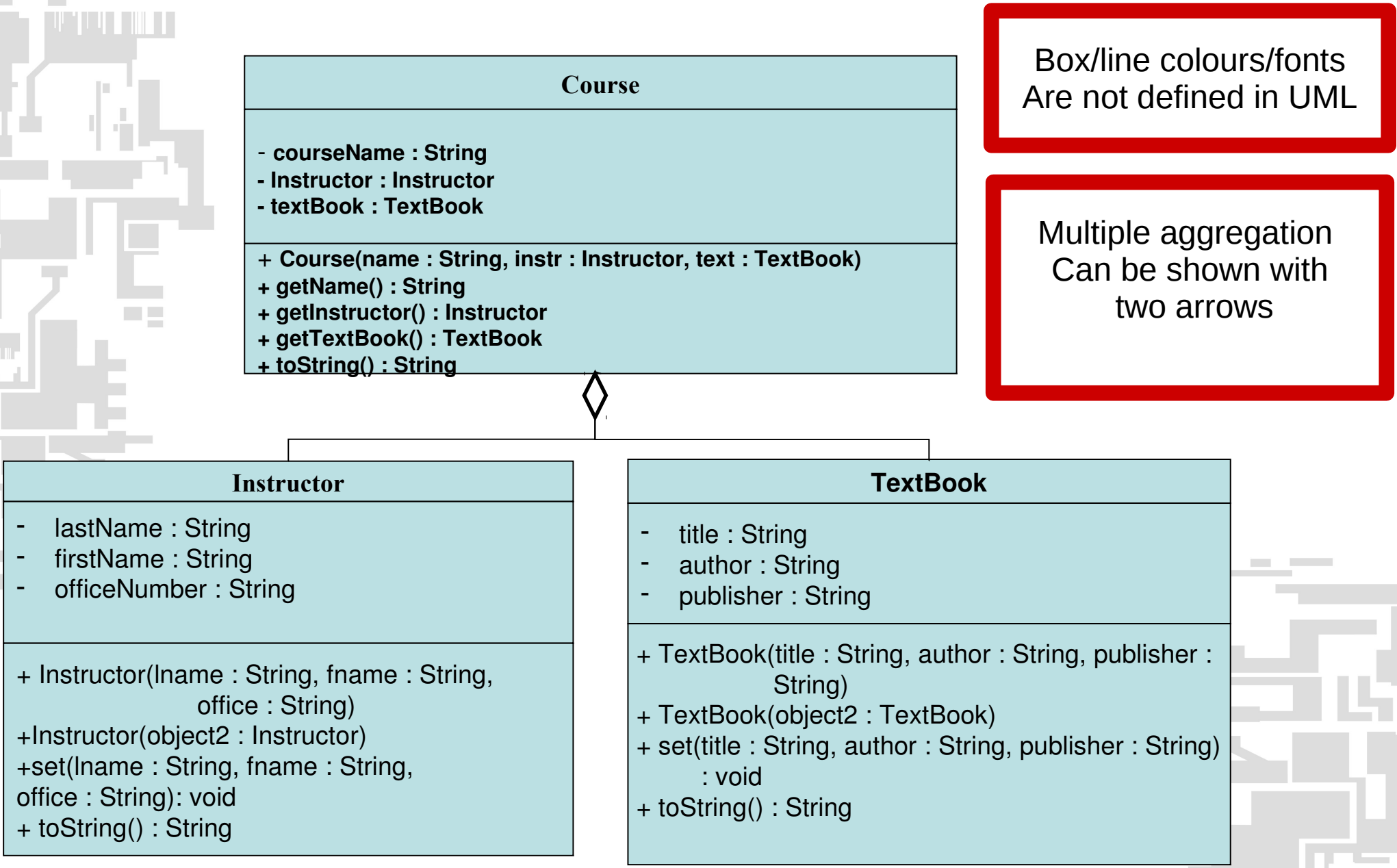
- None of these contain the *implementation* (the code in the method body)
- They only contain the method header:
- The method name
- Any arguments
- The return value
- This is also what a UML Class diagram provides
- Anyone looking at the class API is not interested in the underlying code, only what it can be used for
- Do you need to know what the System.out.println() method does internally when you call it?

Inheritance in UML

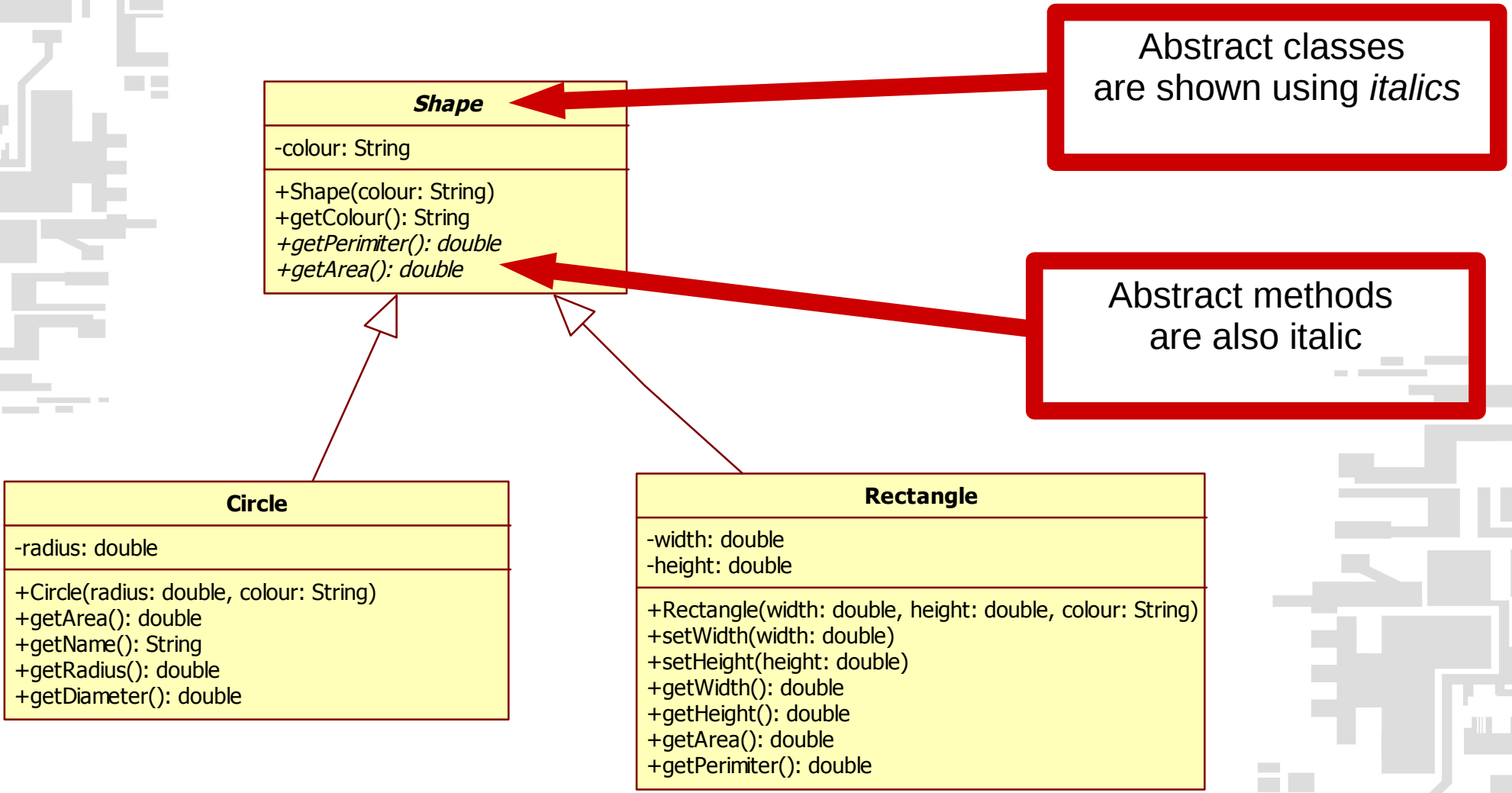


Aggregation




Course
- courseName : String
- Instructor : Instructor
- textBook : TextBook
+ Course(name : String, instr : Instructor, text : TextBook)
+ getName() : String
+ getInstructor() : Instructor
+ getTextBook() : TextBook
+ toString() : String
Instructor
- lastName : String
- firstName : String
- officeNumber : String
+ Instructor(lname : String, fname : String, office : String)
+Instructor(object2 : Instructor)
+set(lname : String, fname : String, office : String): void
+ toString() : String
TextBook
- title : String
- author : String
- publisher : String
+ TextBook(title : String, author : String, publisher : String)
+ TextBook(object2 : TextBook)
+ set(title : String, author : String, publisher : String) : void
+ toString() : String
Box/line colours/fonts
Are not defined in UML
Multiple aggregation
Can be shown with
two arrows

Abstract Classes



UML - Criticism

- UML is not used by all developers and faces some criticism
  - It's very easy for the diagram to get out of sync with the actual code. This is confusing for developers and reduces the usefulness of the diagrams
  - To overcome this, UML diagrams can be generated from the code.
  - However it's usually quicker to just look at the code!