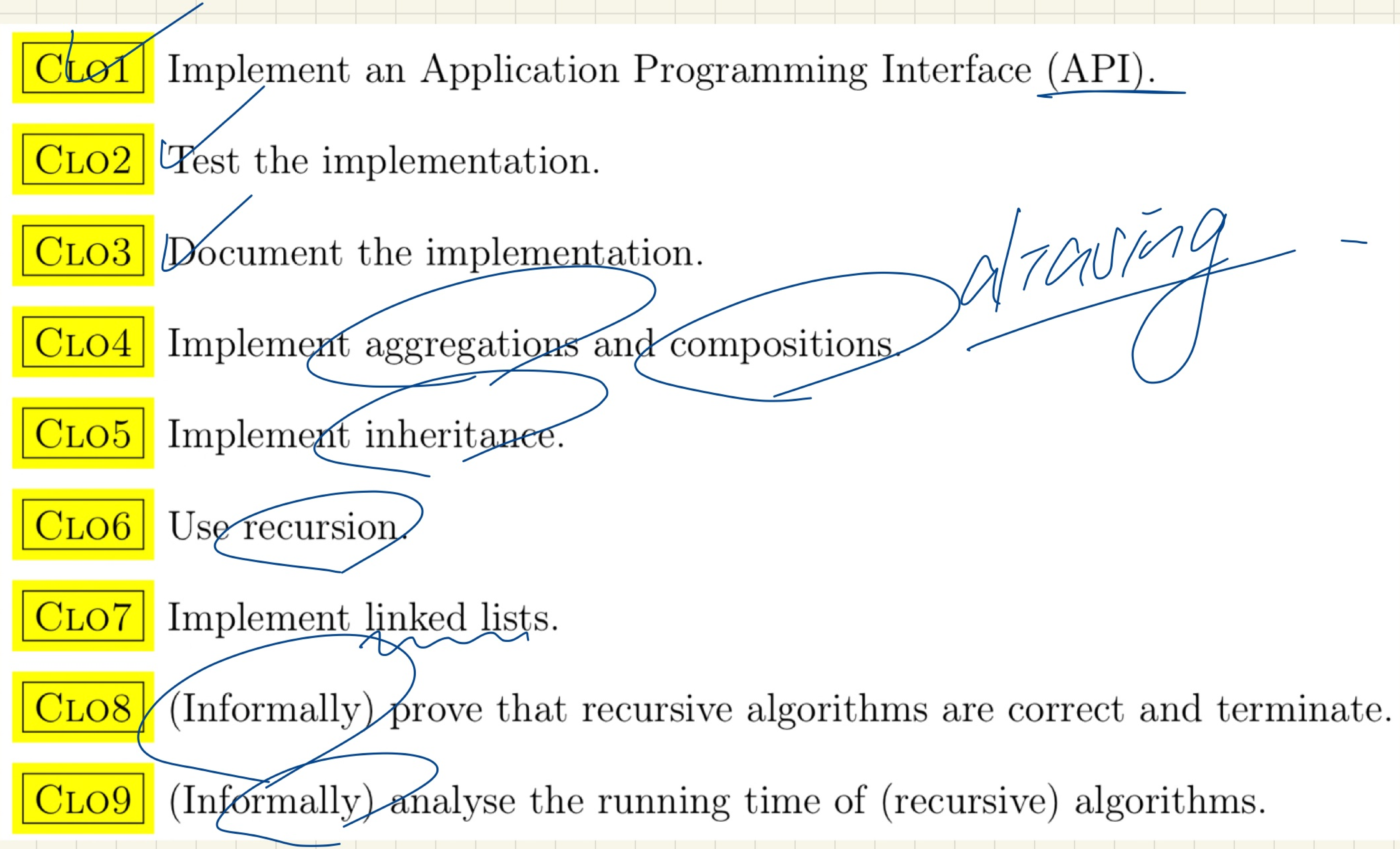
LECTURE 1

Wednesday September 4

COURSE LEARNING OUTCOMES



CLO1 Implement an Application Programming Interface (API).

CLO2 Test the implementation.

CLO3 Document the implementation.

CLO4 Implement aggregations and compositions. drawing -

CLO5 Implement inheritance.

CLO6 Use recursion.

CLO7 Implement linked lists.

CLO8 (Informally) prove that recursive algorithms are correct and terminate.

CLO9 (Informally) analyse the running time of (recursive) algorithms.

Survival Pattern

Lectures
(attention)
15%
Tutorial
Videos
(attention)
15%
Labs
(honesty)
15%
>/30%
< 30%
Recordings
Reflections
45%
Talk to
Jackie
deserved!
bad luck!
SUCCESS
FAILURE

Object-Oriented Programming (OOP)

- Templates (Compile-time Java classes)
  - ~ attributes
  - ~ methods
    - constructor
    - mutator
    - accessor
- Instances / Entities (runtime objects)
  - ~ calling constructor to create objects
  - ~ use of "dot notation" to
    - get attribute values
    - call accessor or mutator

class vs. object

Person

P. getTuition()

P. spouse. spouse

Test Driven Development (TDD)

entry point of execution

tester

```
public class Tester{
  public static void main(String[] args){
    :  /* create and manipulate objects
  }
}
```

later ↳ ~~change to~~ JUnit test class

uses
model
class ...{
Person
:
}
class ...{
Student
:
}
Bad

The Observe-Model-Execute Process

Jim
55
1.81 m
Jonathan
60
1.75 m
jim.getBMI()
jonathan.getBMI()

Real World: Entities

Entities:
jim, jonathan, ...

Entities:
p1(2, 3), p2(-1, -2), ...

...

Model

Compile-Time: Classes
(definitions of templates)

```
class Person {
  String name;
  double weight;
  double height;
}
```

```
class Point {
  double x;
  double y;
}
```

...

Execute

Run-Time: Objects
(instantiations of templates)

jim

Person	
name	"Jim"
weight	80
height	1.80

jonathan

Person	
name	"Jonathan"
weight	80
height	1.80

p1

Point	
x	2
y	3

p2

Point	
x	-1
y	-2

...

context objects

Model: From Entities to Classes

Identify Critical Nouns & Verbs

Example 1

class

A person is a being, such as a human, that has certain attributes and behaviour constituting personhood: a person ages and grows on their heights and weights.

attributes.

Example 2

Points on a two-dimensional plane are identified by their signed distances from the X- and Y-axises. A point may move arbitrarily towards any direction on the plane. Given two points, we are often interested in knowing the distance between them.

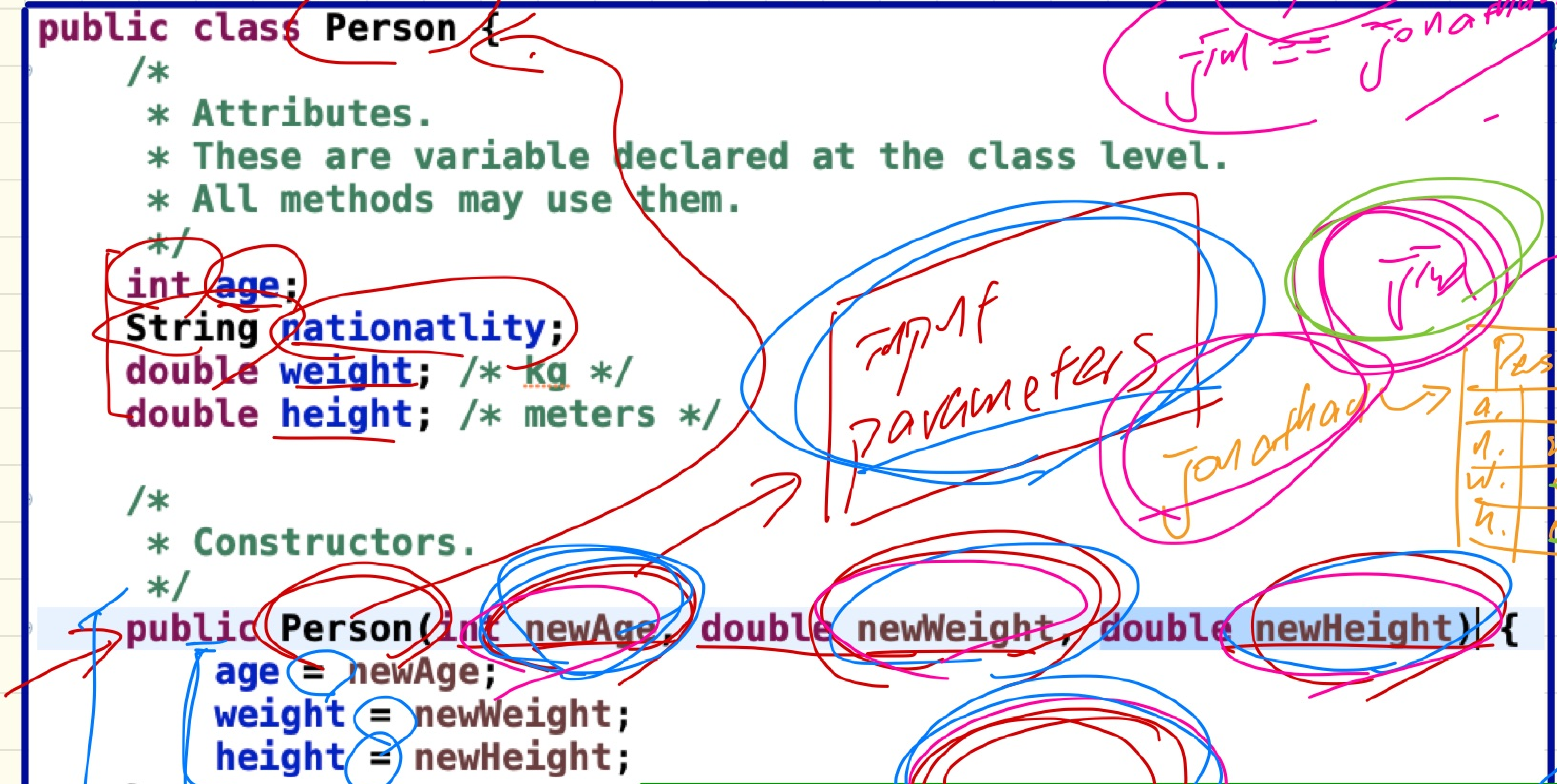
Constructors

Perspectives:

1. Java
2. Debugger

F

jim == jonathan



```java
public class Person {
    /*
     * Attributes.
     * These are variable declared at the class level.
     * All methods may use them.
     */
    int age;
    String nationatlity;
    double weight; /* kg */
    double height; /* meters */

    /*
     * Constructors.
     */
    public Person(int newAge, double newWeight, double newHeight) {
        age = newAge;
        weight = newWeight;
        height = newHeight;
    }
}
```

input parameters

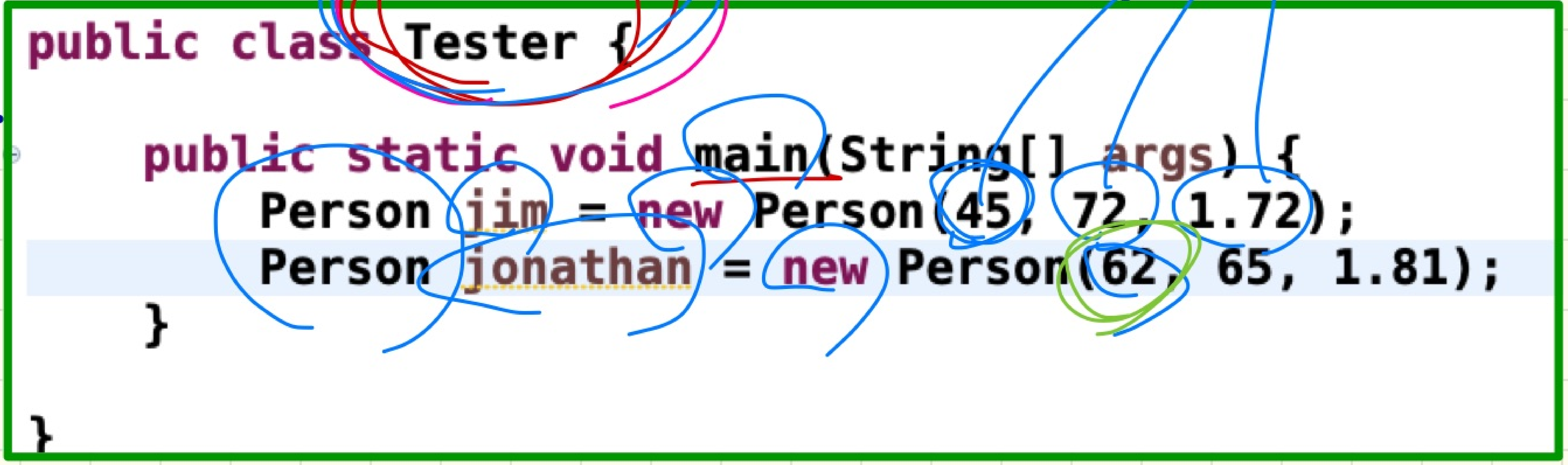
jim

jonathan

Person	
a.	~~0~~ 45
n.	null
w.	~~0.0~~ 72
h.	~~0.0~~

Person	
a.	~~0~~ 62
n.	null
w.	~~0.0~~ 65
h.	~~0.0~~ 1.81

actual arguments



```java
public class Tester {

    public static void main(String[] args) {
        Person jim = new Person(45, 72, 1.72);
        Person jonathan = new Person(62, 65, 1.81);
    }

}
```