

Monday March 4

Lecture 15

# Object-Oriented Programming (OOP)

Templates (compile-time Java classes)

→ ~ attributes variables

~ 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

OOP

## Model: From Entities to Classes

Identify Critical Nouns & Verbs

attributes/classes  
methods

### Example 1

class/template

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.

### Example 2

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

```

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

```

Attributes

| Person |  |
|--------|--|
| age    |  |
| nat.   |  |
| w.     |  |
| h.     |  |

```

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

```

parameters.

definition of constructor

```

public class Tester {
    public static void main() {
        Person jim = new Person(45, 72, 1.72);
    }
}

```

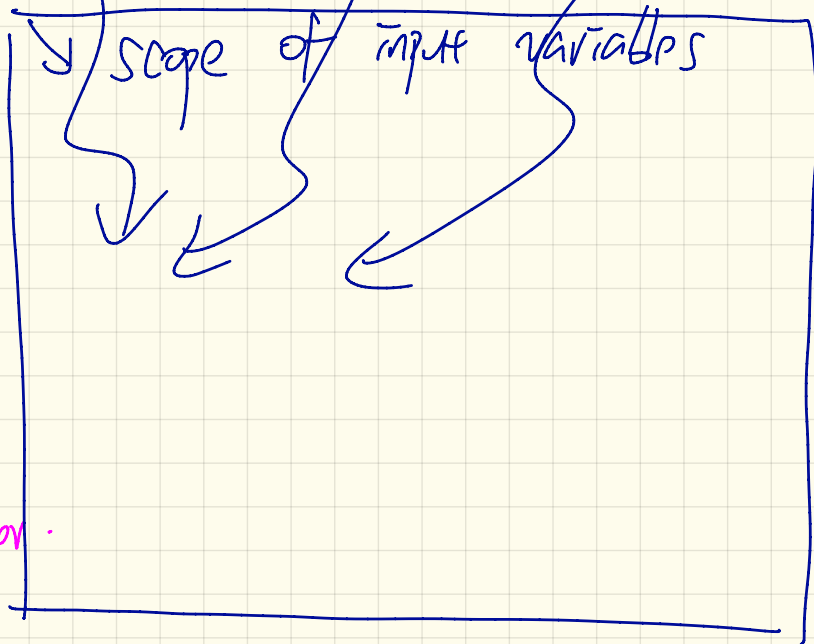
use of constructor

create a new object

Person

name  
of  
constructor

( int newAge double newW double newH ) {



"a list of  
inputs"  
parameters  
for  
the constructor.

{

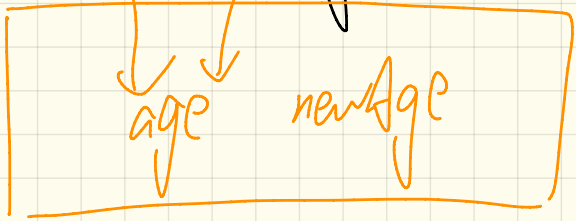
```
class Person {
```

```
    int age;
```

attribut

```
    Person (int newAge, ...)
```

input parameter for this particular constructor



```
}
```

```
}
```

```
public class Person {
    /*
     * Attributes.
     * These are variable declared at the class level.
     * All methods may use them.
     */
```

```
int age;
String nationality;
double weight; /* kg */
double height; /* meters */
```

```
/*
 * Constructors.
 */
```

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

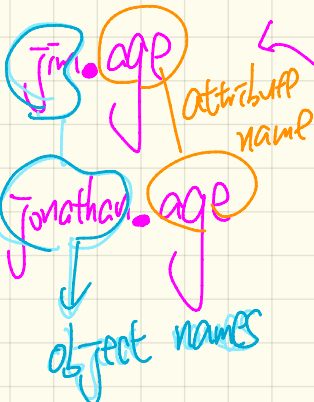
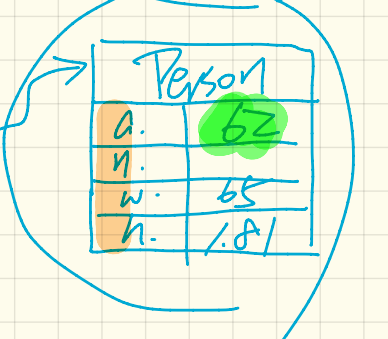
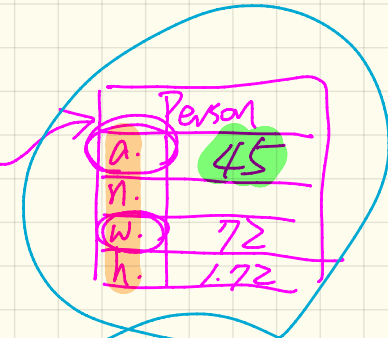
define the constructor

45 62

72 65

1.72 1.81

Jonathan



```
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);
}
```

two calls to constructor

# Test Driven Development (TDD)

testey

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

uses

model

model  
for  
reality

Person

