Programming Fundamentals 1

Produced by

Mr. Dave Drohan (david.drohan@setu.ie)
Dr. Siobhán Drohan
Ms. Mairead Meagher

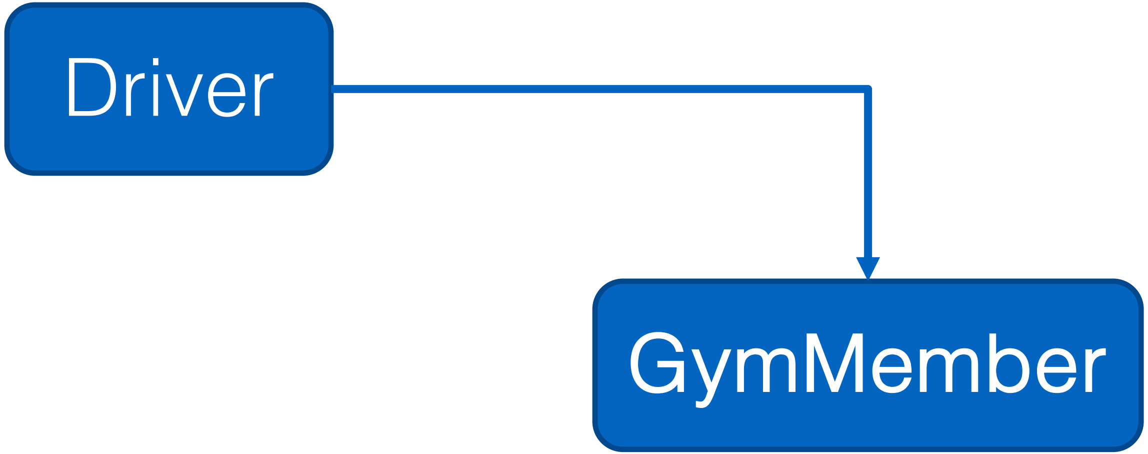
Department of Computing & Mathematics
South East Technological University
Waterford, Ireland

setu.ie

GYM APP

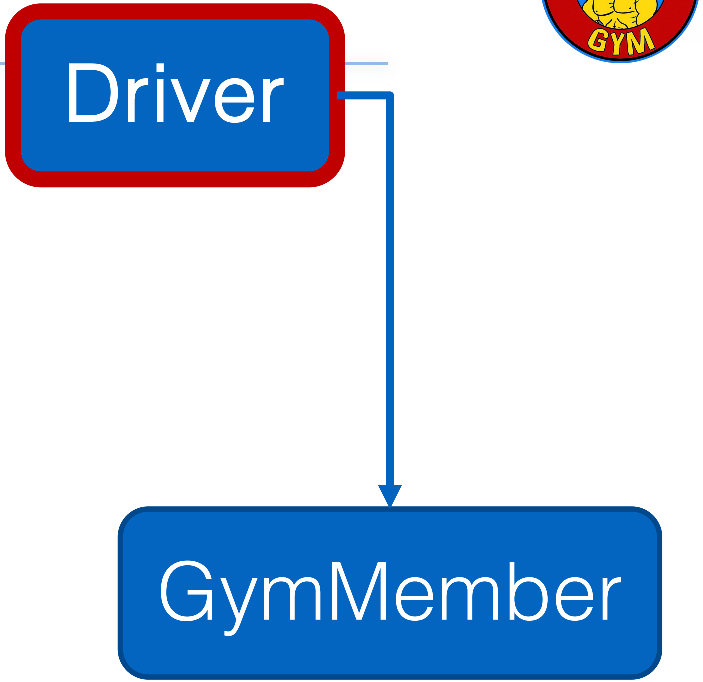
Version 1.0

GYM App V1.0



GYM App V1.0

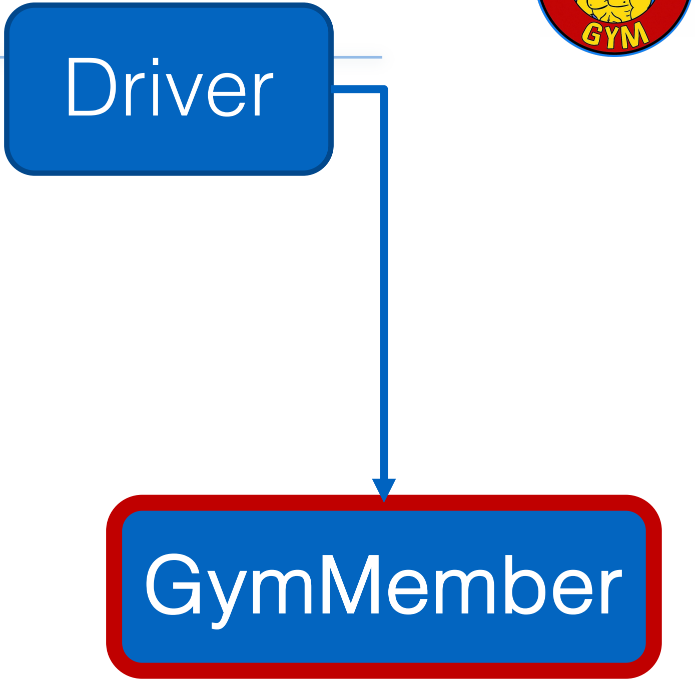
- ❑ The `Driver` class:
  - ❑ has the `main()` method.
  - ❑ reads the gym member details from the user (via the console)
  - ❑ creates a new `GymMember` object.
  - ❑ prints the gym member details from the object (to the console)



GYM App V1.0

- The **GymMember** class stores details about a single member of a gym i.e.:
  - name
  - height
  - weight
  - membershipNumber
  - if they are **current members** or not?

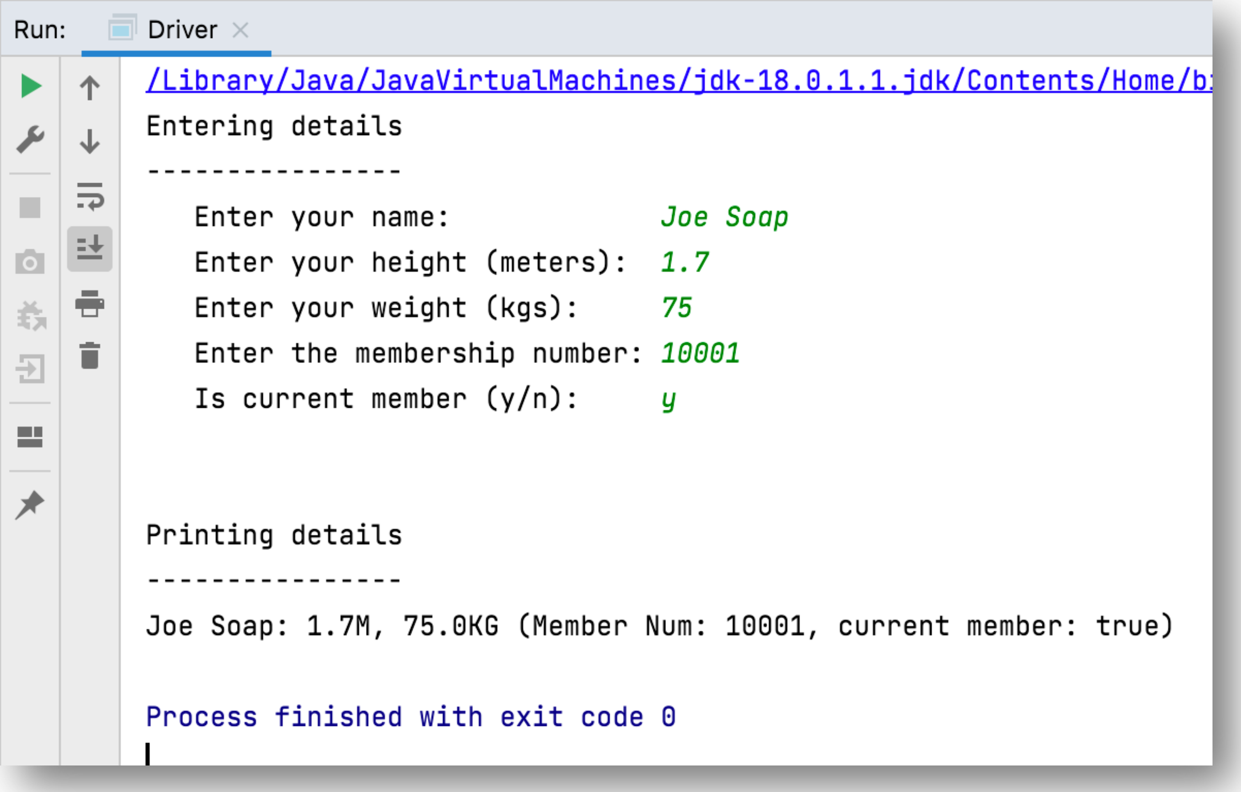
    (i.e. have they paid their membership fee)



Driver

GymMember

GYM App V1.0 – Sample I/O



```
/Library/Java/JavaVirtualMachines/jdk-18.0.1.1.jdk/Contents/Home/b
Entering details
----------------
   Enter your name:              Joe Soap
   Enter your height (meters):   1.7
   Enter your weight (kgs):      75
   Enter the membership number:  10001
   Is current member (y/n):      y


Printing details
----------------
Joe Soap: 1.7M, 75.0KG (Member Num: 10001, current member: true)

Process finished with exit code 0
```

ENCAPSULATED CLASS:

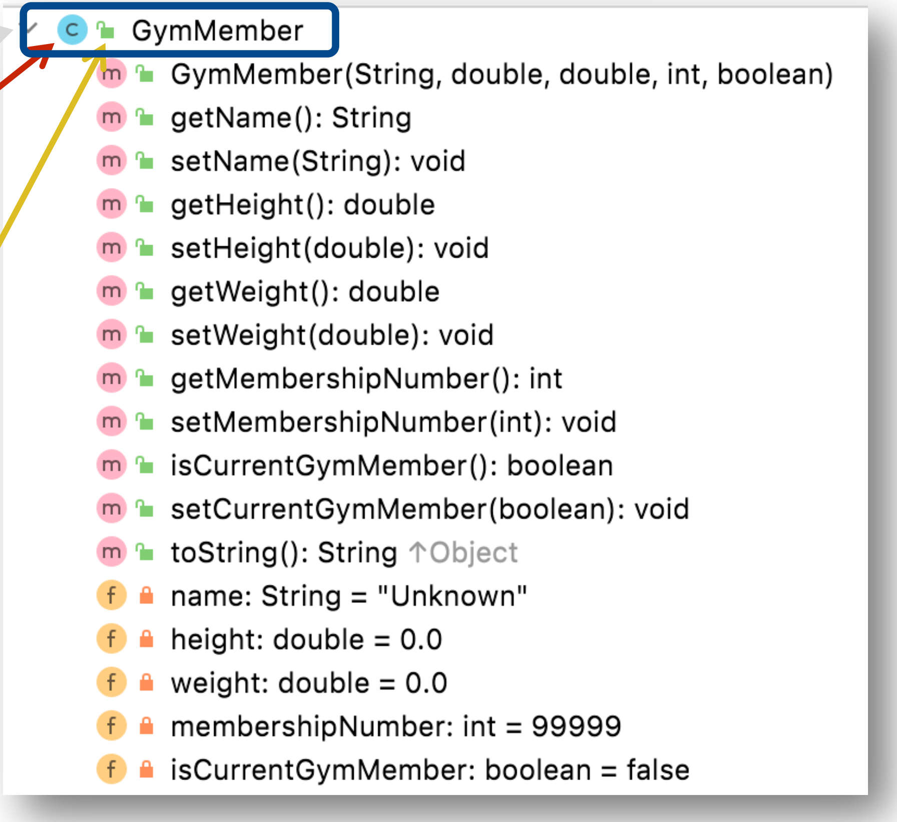
GymMember

A **GymMember** Class…

Object Type / **Class** Name
i.e. **GymMember**

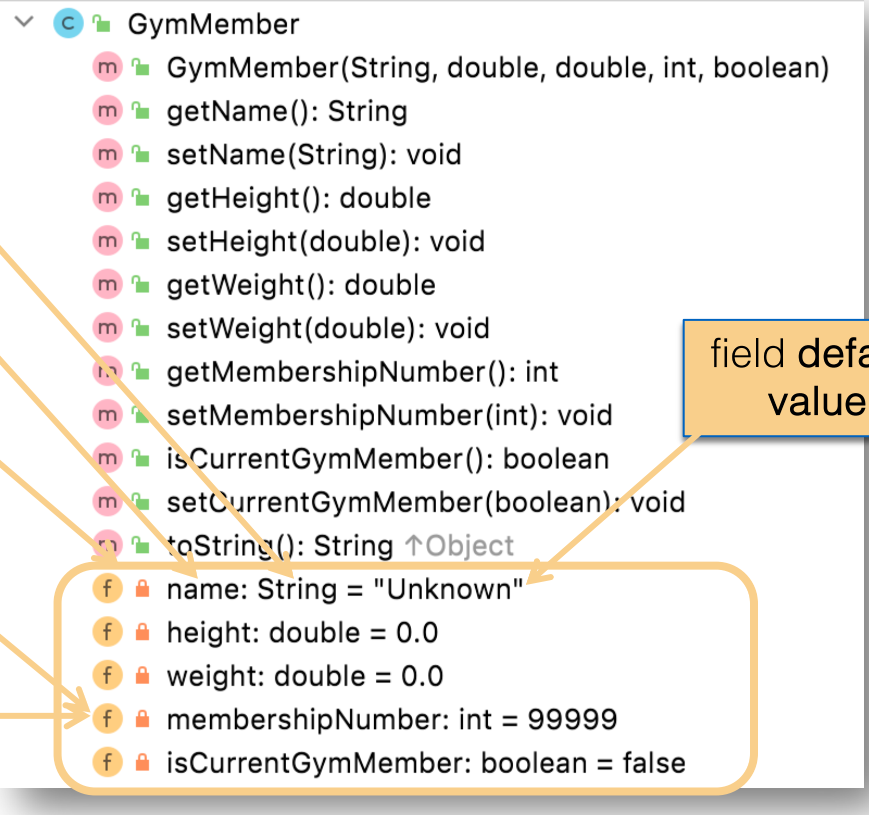
The **C** icon means it is a **Class**.

The open padlock means it is **public**.



```
c GymMember
    m GymMember(String, double, double, int, boolean)
    m getName(): String
    m setName(String): void
    m getHeight(): double
    m setHeight(double): void
    m getWeight(): double
    m setWeight(double): void
    m getMembershipNumber(): int
    m setMembershipNumber(int): void
    m isCurrentGymMember(): boolean
    m setCurrentGymMember(boolean): void
    m toString(): String ↑Object
    f name: String = "Unknown"
    f height: double = 0.0
    f weight: double = 0.0
    f membershipNumber: int = 99999
    f isCurrentGymMember: boolean = false
```

A **GymMember** Class…



```
c GymMember
    m GymMember(String, double, double, int, boolean)
    m getName(): String
    m setName(String): void
    m getHeight(): double
    m setHeight(double): void
    m getWeight(): double
    m setWeight(double): void
    m getMembershipNumber(): int
    m setMembershipNumber(int): void
    m isCurrentGymMember(): boolean
    m setCurrentGymMember(boolean): void
    m toString(): String ↑Object
    f name: String = "Unknown"
    f height: double = 0.0
    f weight: double = 0.0
    f membershipNumber: int = 99999
    f isCurrentGymMember: boolean = false
```

field **type**

field **name**

The closed padlock means it is **private**.

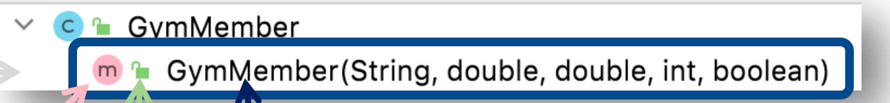
The **f** icon means it is a **field**.

Fields
i.e. the **attributes / properties** of the class

field **default value**

A `GymMember` Class… constructor

Constructor
i.e. for building objects.



GymMember

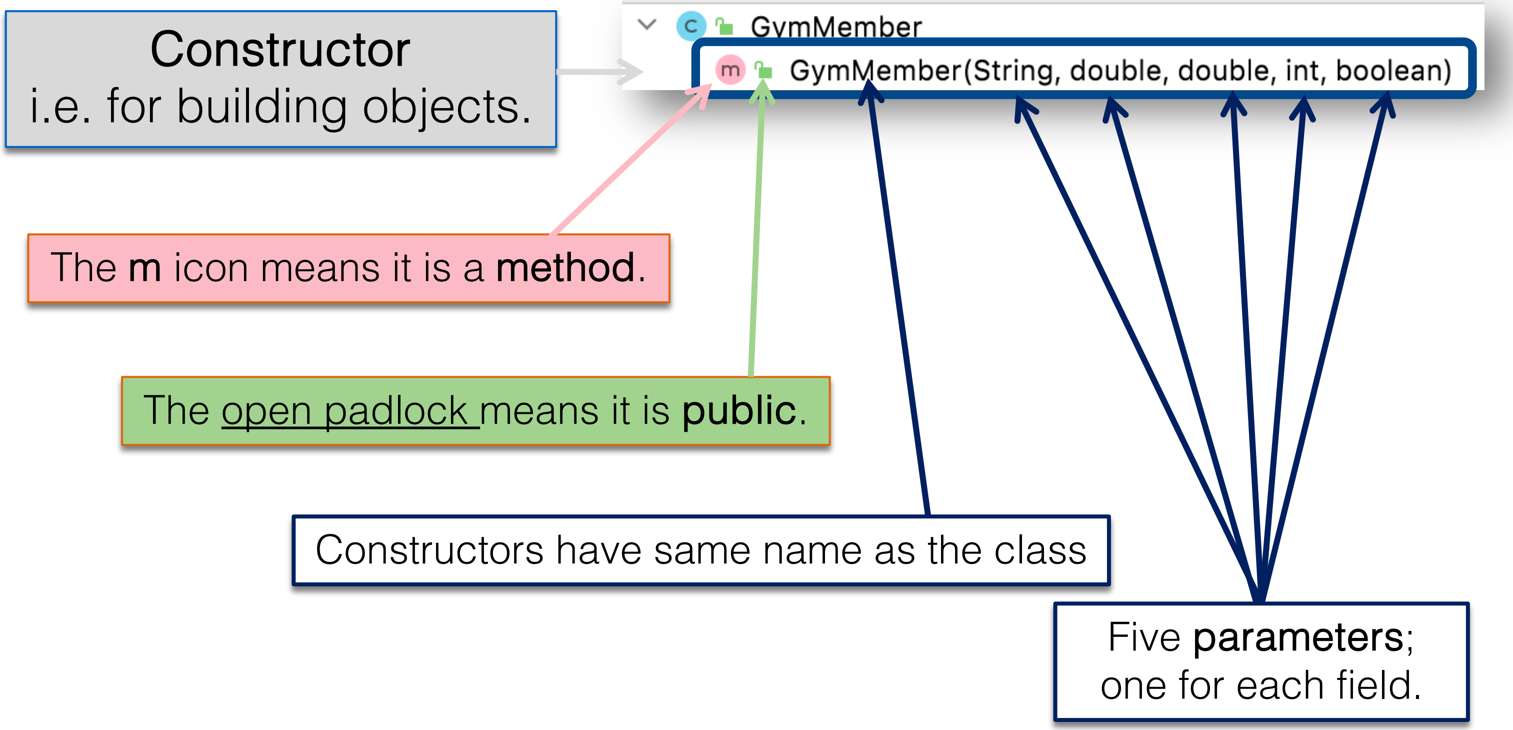
GymMember(String, double, double, int, boolean)

The **m** icon means it is a **method**.

The open padlock means it is **public**.

Constructors have same name as the class

A GymMember Class… constructor



GymMember

GymMember(String, double, double, int, boolean)

Constructor
i.e. for building objects.

The **m** icon means it is a **method**.

The open padlock means it is **public**.

Constructors have same name as the class

Five **parameters**;
one for each field.

A **GymMember** Class… Fields & Constructor

```java
private String name = "Unknown";
private double height = 0.0
private double weight = 0.0;
private int membershipNumber = 99999;
private boolean isCurrentGymMember = false;
```

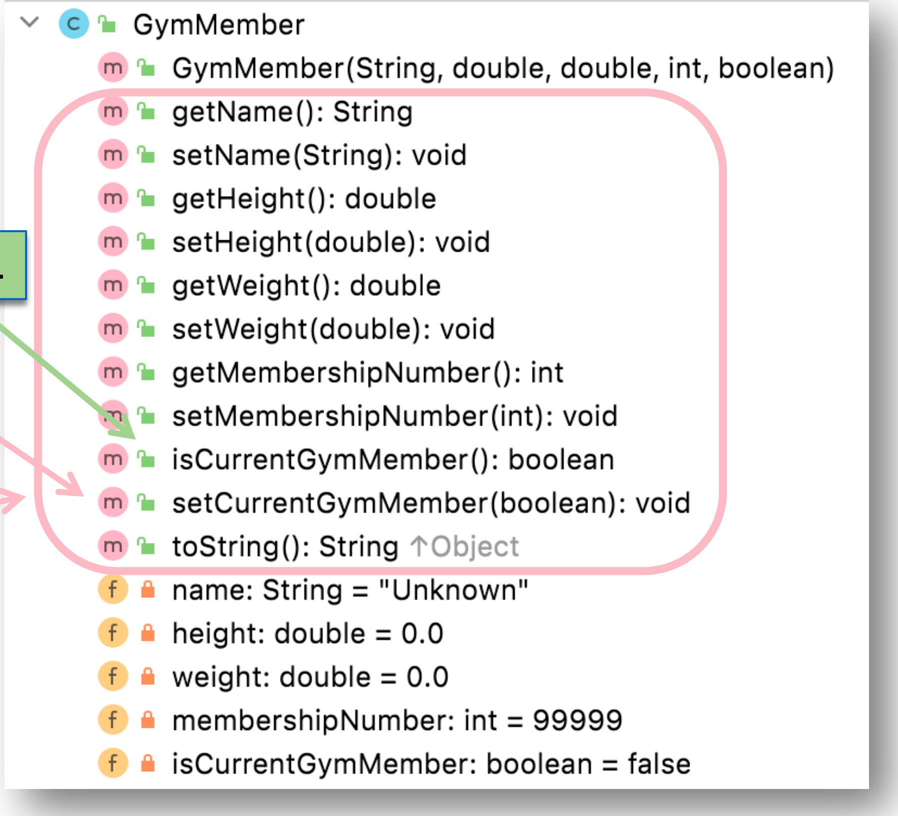
```java
public GymMember(String name, double height, double weight, int
membershipNumber, boolean isCurrentGymMember) {
    setName(name);
    setHeight(height);
    setWeight(weight);
    setMembershipNumber(membershipNumber);
    setCurrentGymMember(isCurrentGymMember);
}
```

A GymMember Class… methods

The open padlock means it is **public**.

The **m** icon means it is a **method**.

Methods
i.e. the **behaviours** of the class

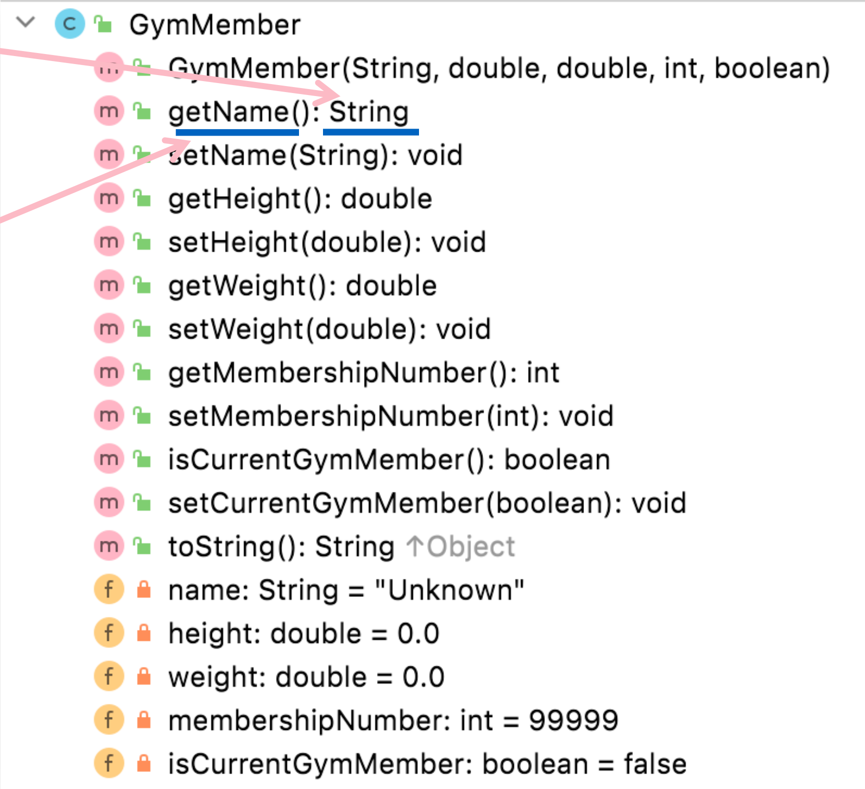


```
GymMember
    GymMember(String, double, double, int, boolean)
    getName(): String
    setName(String): void
    getHeight(): double
    setHeight(double): void
    getWeight(): double
    setWeight(double): void
    getMembershipNumber(): int
    setMembershipNumber(int): void
    isCurrentGymMember(): boolean
    setCurrentGymMember(boolean): void
    toString(): String ↑Object
    name: String = "Unknown"
    height: double = 0.0
    weight: double = 0.0
    membershipNumber: int = 99999
    isCurrentGymMember: boolean = false
```

A **GymMember** Class… methods

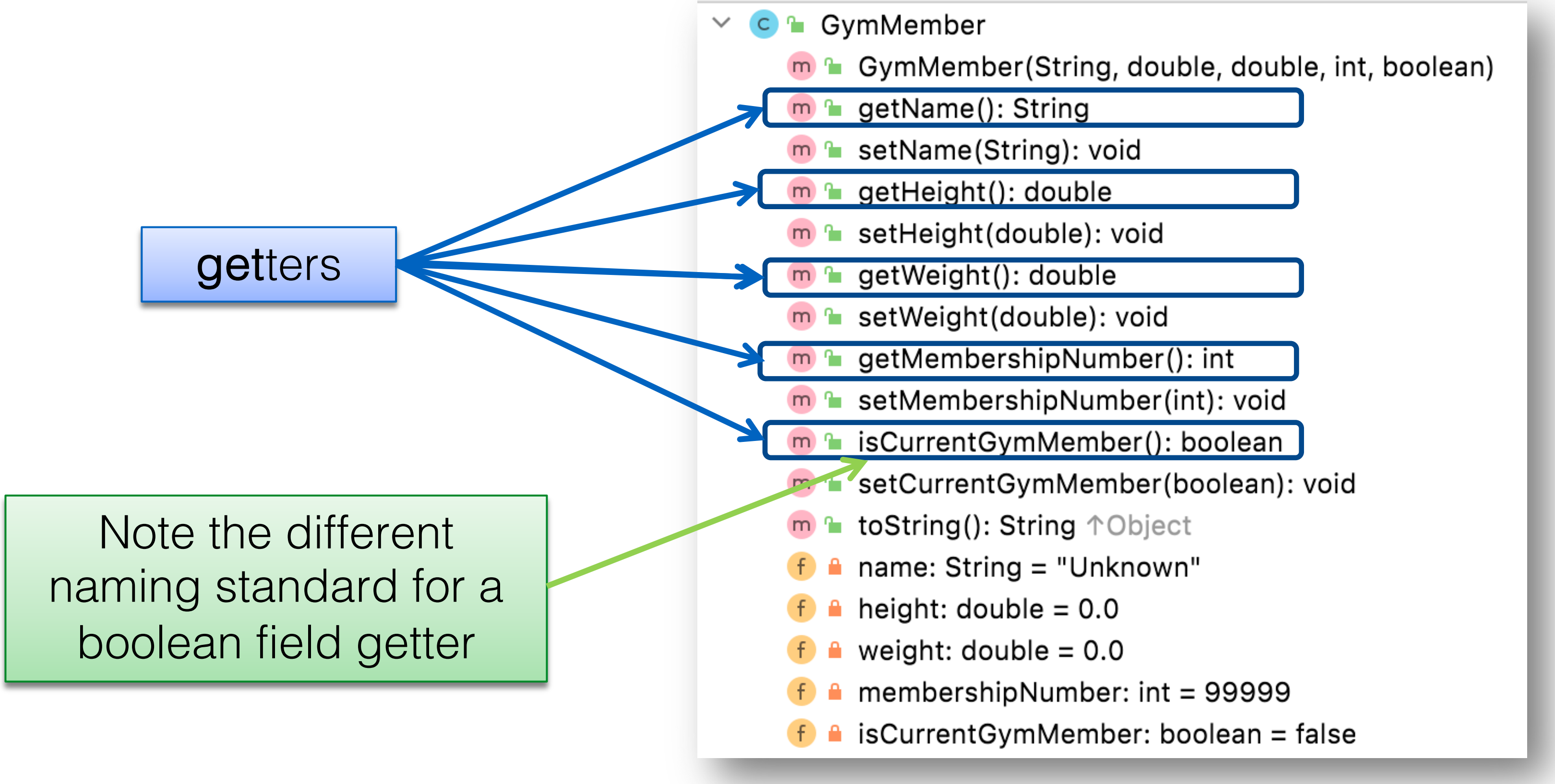
Return type

Method name



```
GymMember
    GymMember(String, double, double, int, boolean)
    getName(): String
    setName(String): void
    getHeight(): double
    setHeight(double): void
    getWeight(): double
    setWeight(double): void
    getMembershipNumber(): int
    setMembershipNumber(int): void
    isCurrentGymMember(): boolean
    setCurrentGymMember(boolean): void
    toString(): String ↑Object
    name: String = "Unknown"
    height: double = 0.0
    weight: double = 0.0
    membershipNumber: int = 99999
    isCurrentGymMember: boolean = false
```

A **GymMember** Class… getters



getters

Note the different naming standard for a boolean field getter

```
GymMember
  GymMember(String, double, double, int, boolean)
  getName(): String
  setName(String): void
  getHeight(): double
  setHeight(double): void
  getWeight(): double
  setWeight(double): void
  getMembershipNumber(): int
  setMembershipNumber(int): void
  isCurrentGymMember(): boolean
  setCurrentGymMember(boolean): void
  toString(): String ↑Object
  name: String = "Unknown"
  height: double = 0.0
  weight: double = 0.0
  membershipNumber: int = 99999
  isCurrentGymMember: boolean = false
```

Getters (Accessor Methods)

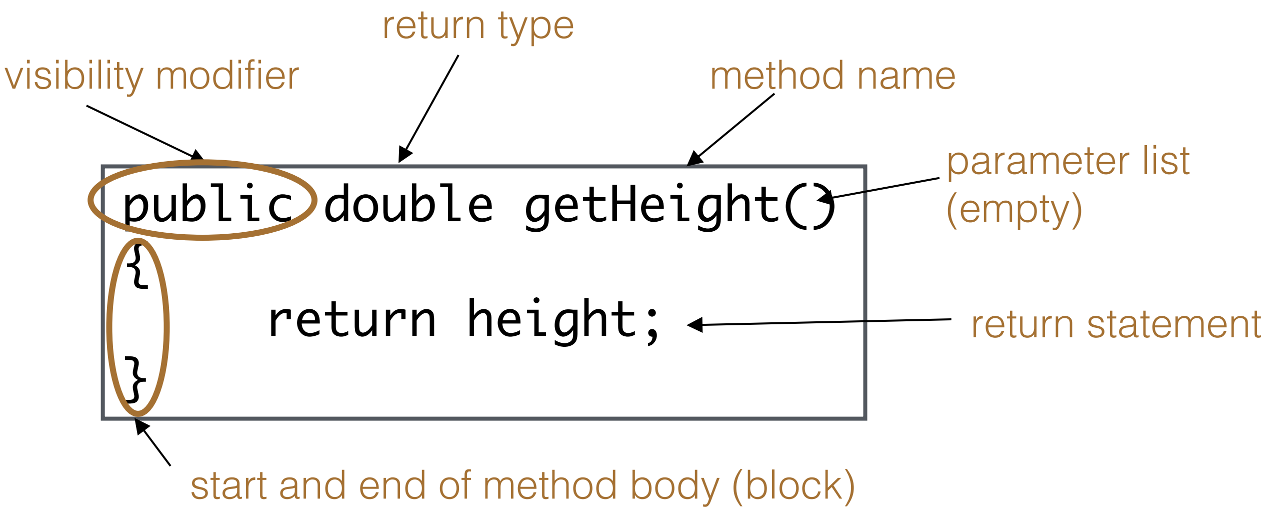
❑ Accessor methods

- return information about the **state** of an object
  - i.e. **the values stored in the fields.**

❑ A ‘getter’ method

- is a specific type of accessor method and typically:
  - **contains a return statement** (as the last executable statement in the method).
  - defines a **return type.**
  - **does NOT change the object state.**

Getters



return type

visibility modifier

method name

```
public double getHeight()
{
    return height;
}
```

parameter list
(empty)

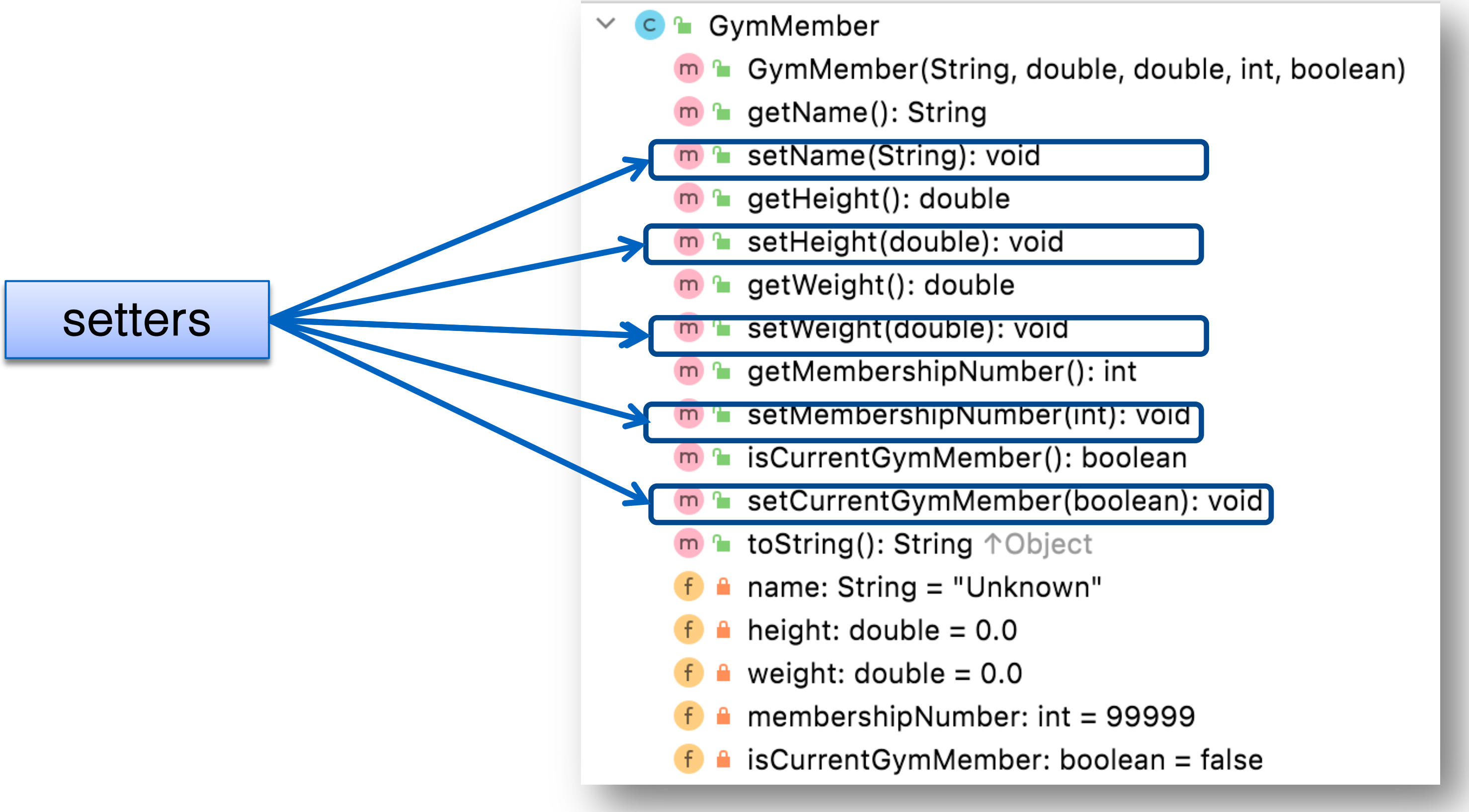
return statement

start and end of method body (block)

A **GymMember** Class… getters

```
public String getName() {
    return name;
}

public double getHeight() {
    return height;
}

public double getWeight() {
    return weight;
}


public int getMembershipNumber()
{
    return membershipNumber;
}

public boolean
isCurrentGymMember() {
    return isCurrentGymMember;
}
```

A **GymMember** Class… setters



setters

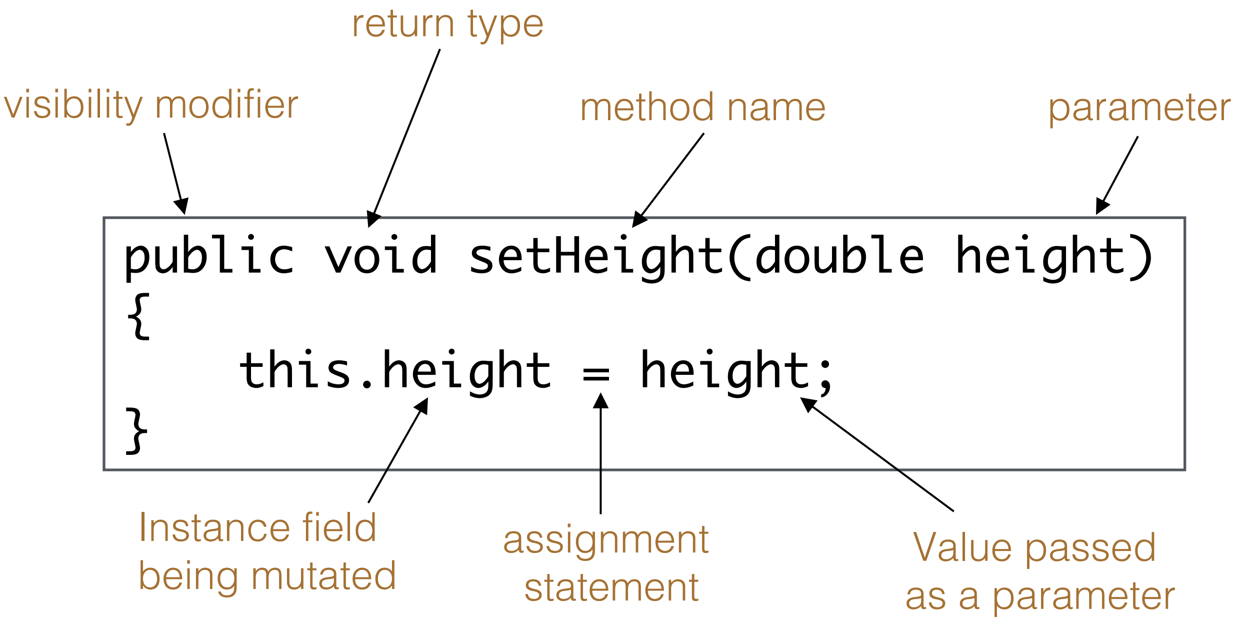
```
GymMember
    GymMember(String, double, double, int, boolean)
    getName(): String
    setName(String): void
    getHeight(): double
    setHeight(double): void
    getWeight(): double
    setWeight(double): void
    getMembershipNumber(): int
    setMembershipNumber(int): void
    isCurrentGymMember(): boolean
    setCurrentGymMember(boolean): void
    toString(): String ↑Object
    name: String = "Unknown"
    height: double = 0.0
    weight: double = 0.0
    membershipNumber: int = 99999
    isCurrentGymMember: boolean = false
```

Setters (Mutator methods)

- Mutator methods
  - change (i.e. mutate!) an object’s state.

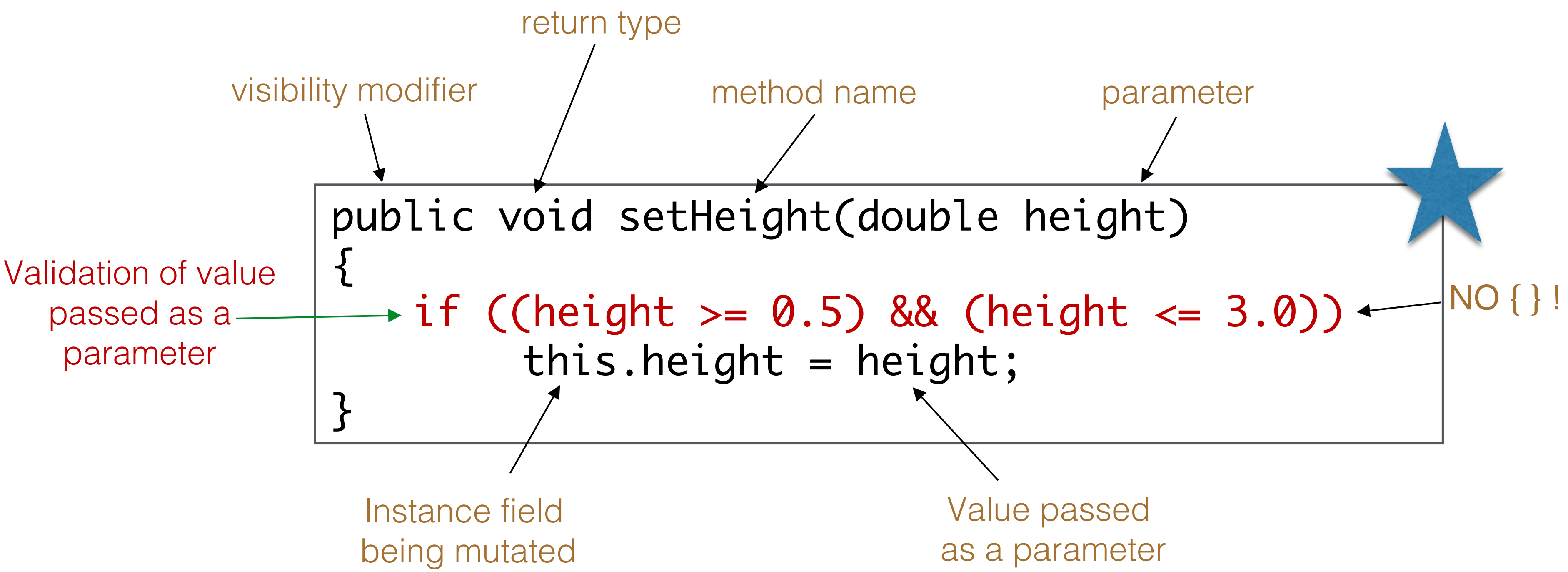
- A ‘setter’ method
  - is a specific type of mutator method and typically:
    - contains an **assignment statement**
    - takes in a **parameter**
    - **changes the object state.**

Setters – without validation



```
public void setHeight(double height)
{
     this.height = height;
}
```

Setters – with validation



return type

visibility modifier

method name

parameter

```
public void setHeight(double height)
{
    if ((height >= 0.5) && (height <= 3.0))
          this.height = height;
}
```

Validation of value passed as a parameter

NO { } !

Instance field being mutated

Value passed as a parameter

A GymMember Class… setters

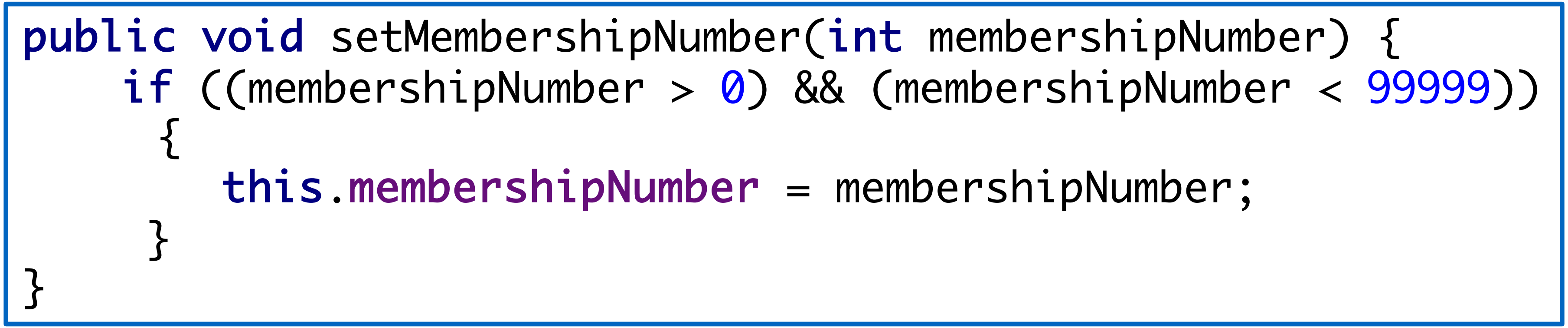
```java
public void setName(String name) {
    if (name != null){
        if (name.length() > 30)
            this.name = name.substring(0,30);
        else
            this.name = name;
        }
}
```

A **GymMember** Class… setters

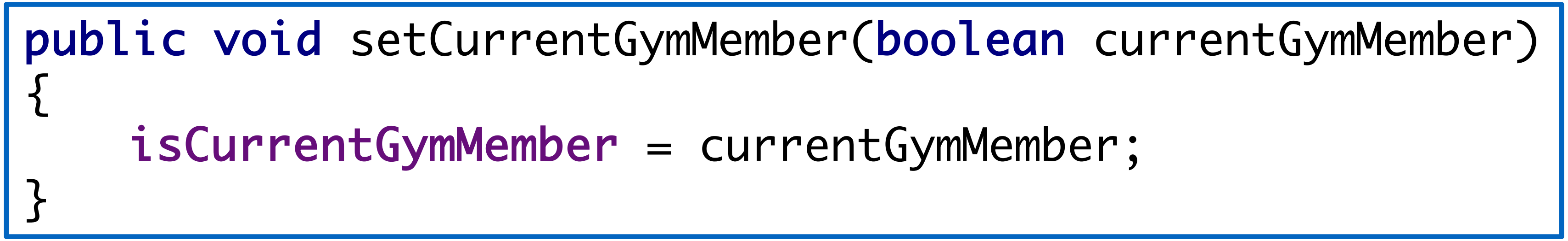
```java
public void setHeight(double height) {
    if ((height >= 0.5) && (height <= 3.0)) {
        this.height = height;
    }
}
```

```java
public void setWeight(double weight) {
    if ((weight >= 25) && (weight <= 500)) {
        this.weight = weight;
    }
}
```

A **GymMember** Class… setters



```
public void setMembershipNumber(int membershipNumber) {
    if ((membershipNumber > 0) && (membershipNumber < 99999))
     {
        this.membershipNumber = membershipNumber;
     }
}
```



```
public void setCurrentGymMember(boolean currentGymMember)
{
    isCurrentGymMember = currentGymMember;
}
```

Getters / Setters

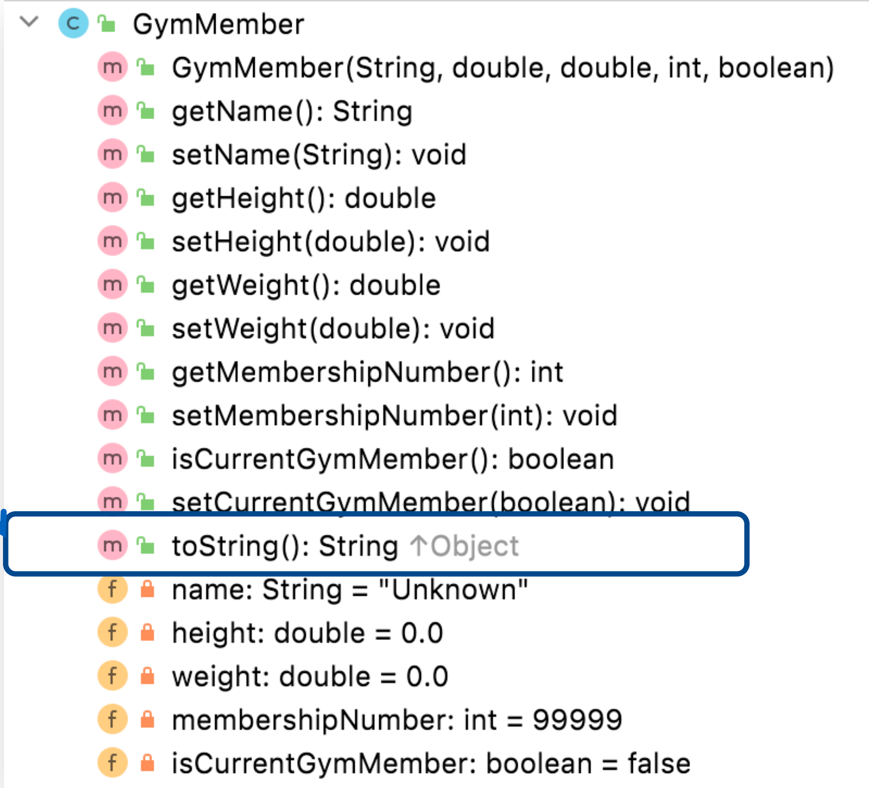
For each instance field in a class, you are normally asked to write:

- A **get**ter
  - Return statement
- A **set**ter
  - Assignment statement

A GymMember Class… toString

toString():

Builds and returns a String containing a user-friendly representation of the object state.

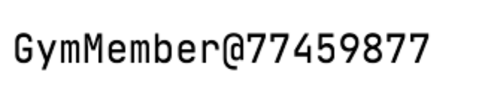


```
GymMember
  GymMember(String, double, double, int, boolean)
  getName(): String
  setName(String): void
  getHeight(): double
  setHeight(double): void
  getWeight(): double
  setWeight(double): void
  getMembershipNumber(): int
  setMembershipNumber(int): void
  isCurrentGymMember(): boolean
  setCurrentGymMember(boolean): void
  toString(): String ↑Object
  name: String = "Unknown"
  height: double = 0.0
  weight: double = 0.0
  membershipNumber: int = 99999
  isCurrentGymMember: boolean = false
```

A GymMember Class… toString

```java
@Override
public String toString() {
    return  name + ": " +
      height + "M, " +
      weight + "KG (Member Num: " +
      membershipNumber + ", current member: " + isCurrentGymMember + ")";
}
```

Sample Console Output if we don't have a toString for GymMember:



```
GymMember@77459877
```

toString()

- This is a useful method and you will write a **toString()** method for most of your classes
- When you print an object, Java automatically calls the **toString()** method e.g.

```
GymMember gymMember = new GymMember();

//both of these lines of code do the same thing
System.out.println(gymMember);
System.out.println(gymMember.toString());
```

USER I/O:

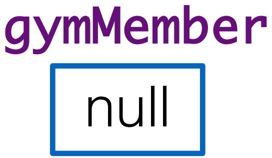
Driver

Using the `GymMember` Class

1

```
private GymMember gymMember;
```

Declaring an object **gymMember**, of type GymMember



gymMember

null

Using the GymMember Class

1

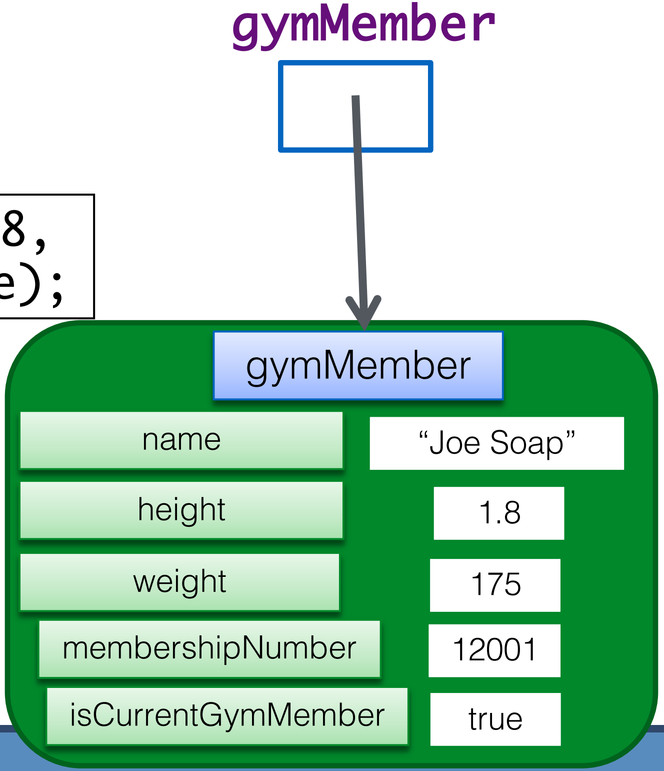
```
private GymMember gymMember;
```

Declaring an object **gymMember**, of type GymMember

2

```
gymMember = new GymMember(“Joe Soap”, 1.8,
                          175, 12001, true);
```

Calls the GymMember *constructor* to build the **gymMember** object in memory.



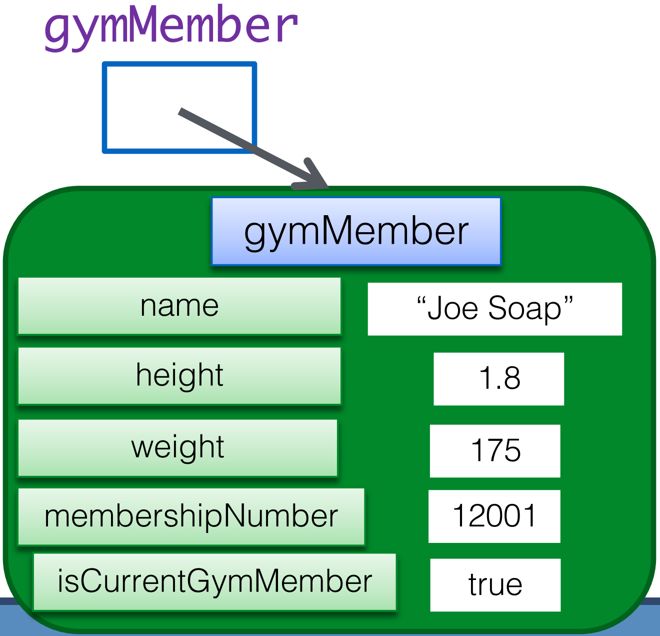
gymMember

gymMember

name	“Joe Soap”
height	1.8
weight	175
membershipNumber	12001
isCurrentGymMember	true

Multiple GymMember objects

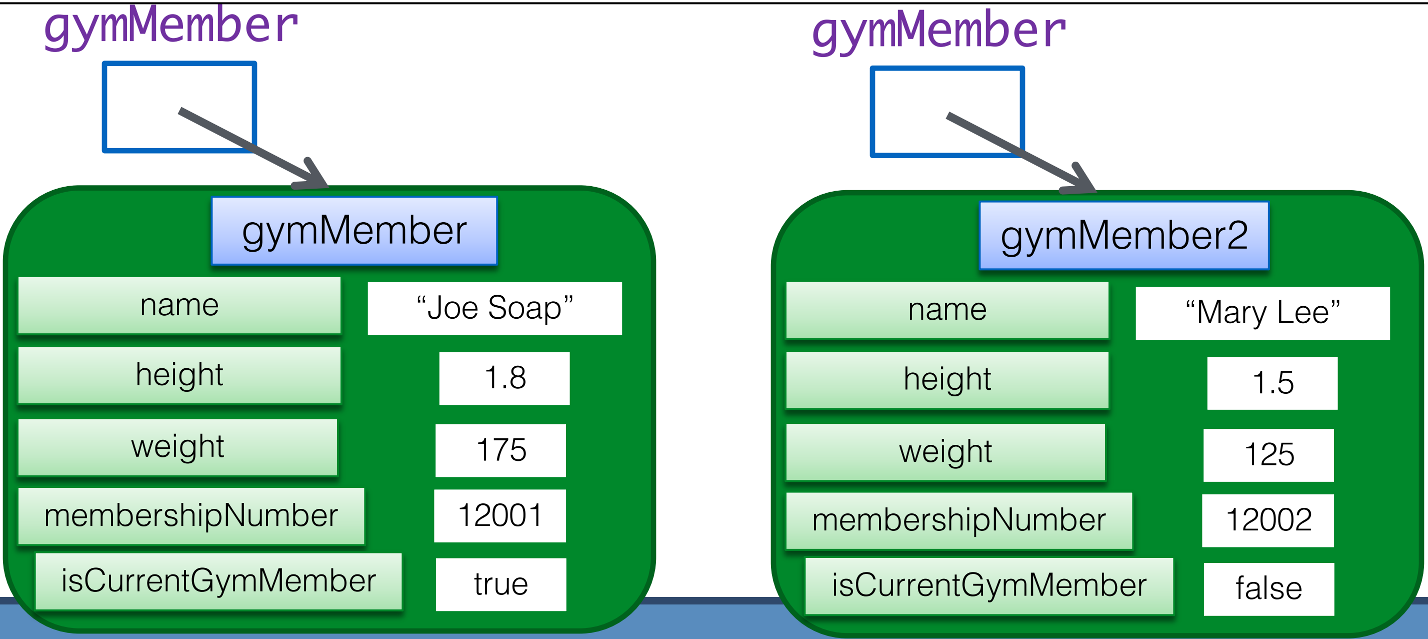
```
private GymMember gymMember = new GymMember("Joe Soap", 1.8,
                                            175, 12001, true);
```



Multiple GymMember objects

```
private GymMember gymMember = new GymMember(“Joe Soap”, 1.8,
                                            175, 12001, true);
```

```
private GymMember gymMember2 = new GymMember(“Mary Lee”, 1.5,
                                             125, 12002, false);
```



gymMember

gymMember	
name	“Joe Soap”
height	1.8
weight	175
membershipNumber	12001
isCurrentGymMember	true

gymMember

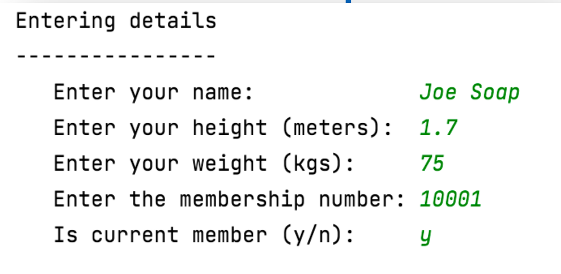
gymMember2	
name	“Mary Lee”
height	1.5
weight	125
membershipNumber	12002
isCurrentGymMember	false

Driver Class

```java
import java.util.Scanner;

public class Driver {

        private Scanner input = new Scanner(System.in);
        private GymMember gymMember;

        public static void main(String[] arg) {
            new Driver();
        }

        public Driver() {
            addGymMember();
            printGymMember();
        }

        // addGymMember () code
        // printGymMember() code

}
```

Driver Class

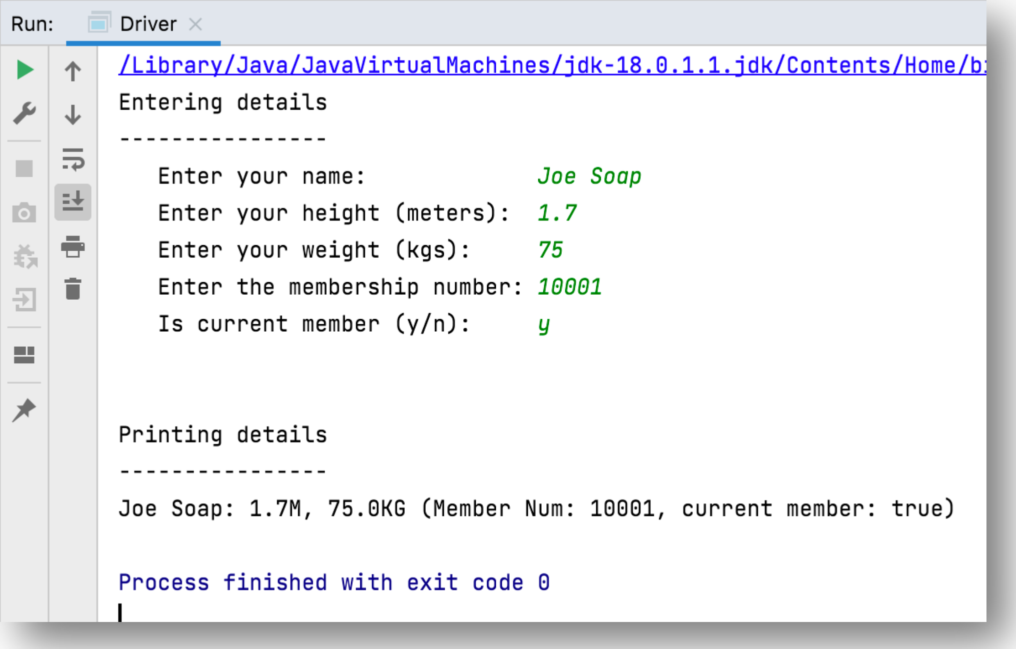
```
private void addGymMember() {
    //obtaining the data from the user
    System.out.println("Entering details");
    System.out.println("----------------");
    System.out.print("   Enter your name:             ");
    String name = input.nextLine();
    System.out.print("   Enter your height (meters):  ");
    double height = input.nextDouble();
    System.out.print("   Enter your weight (kgs):     ");
    double weight = input.nextDouble();
    System.out.print("   Enter the membership number: ");
    int membershipNumber = input.nextInt();
    System.out.print("   Is current member (y/n):     ");
    char isCurrentMemberChar = input.next().charAt(0);

    boolean isCurrentMember = false;
    if ((isCurrentMemberChar == 'Y') || (isCurrentMemberChar == 'y')) {
        isCurrentMember = true;
    }

    gymMember = new GymMember(name, height, weight, membershipNumber,
isCurrentMember);
}
```



```
Entering details
----------------
   Enter your name:             Joe Soap
   Enter your height (meters):  1.7
   Enter your weight (kgs):     75
   Enter the membership number: 10001
   Is current member (y/n):     y
```

Driver Class

```java
private void printGymMember()
{
    //printing out the data to the user
    System.out.println("\n\nPrinting details");
    System.out.println("----------------");
    System.out.println(gymMember);
}
```



```
/Library/Java/JavaVirtualMachines/jdk-18.0.1.1.jdk/Contents/Home/b
Entering details
----------------
    Enter your name:              Joe Soap
    Enter your height (meters):   1.7
    Enter your weight (kgs):      75
    Enter the membership number:  10001
    Is current member (y/n):      y



Printing details
----------------
Joe Soap: 1.7M, 75.0KG (Member Num: 10001, current member: true)

Process finished with exit code 0
```

Questions?

Thanks.