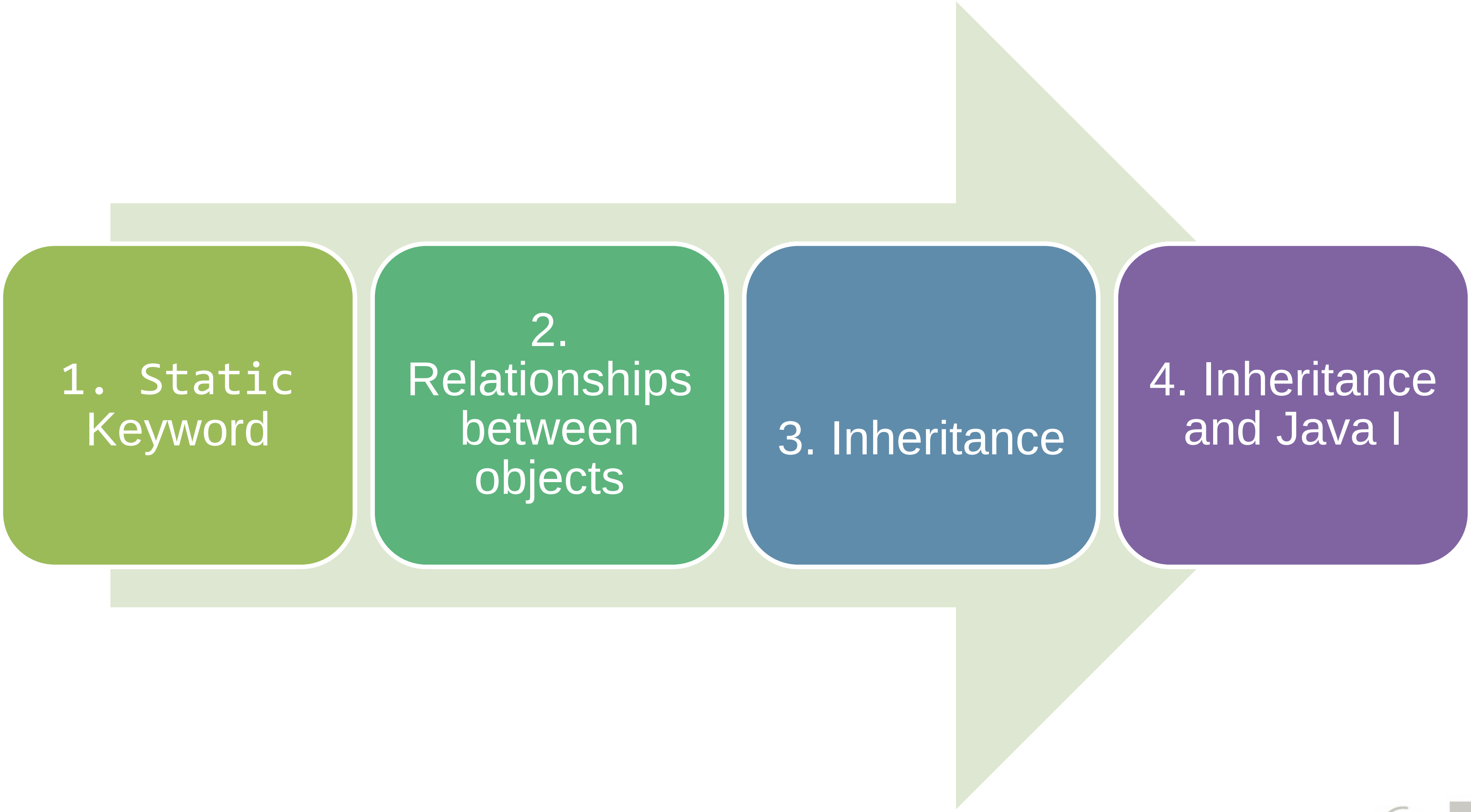
Keyword static

Christian Rodríguez Bustos

Edited by Juan Mendivelso

Object Oriented Programming

Agenda



1. Static Keyword

2. Relationships between objects

3. Inheritance

4. Inheritance and Java I

1. static Keyword

1.1 Static attributes

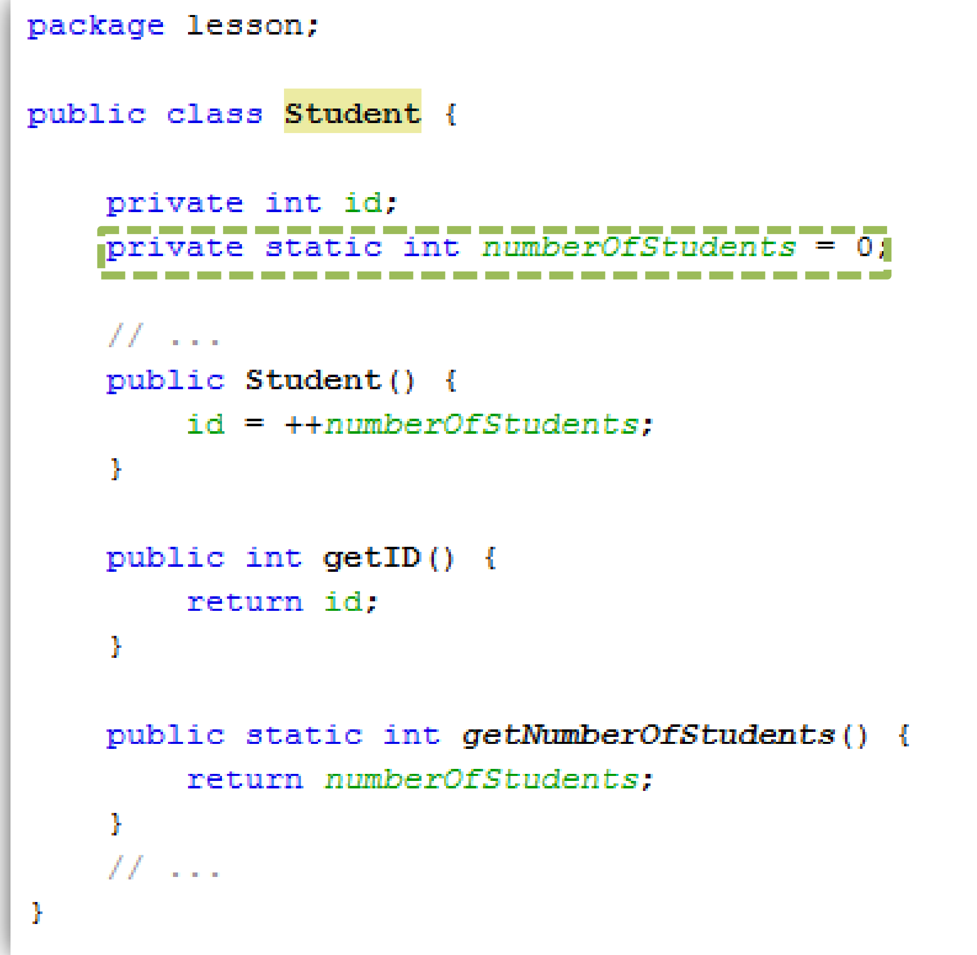
1.2 Static methods

1.1 Static Attributes

Static attributes

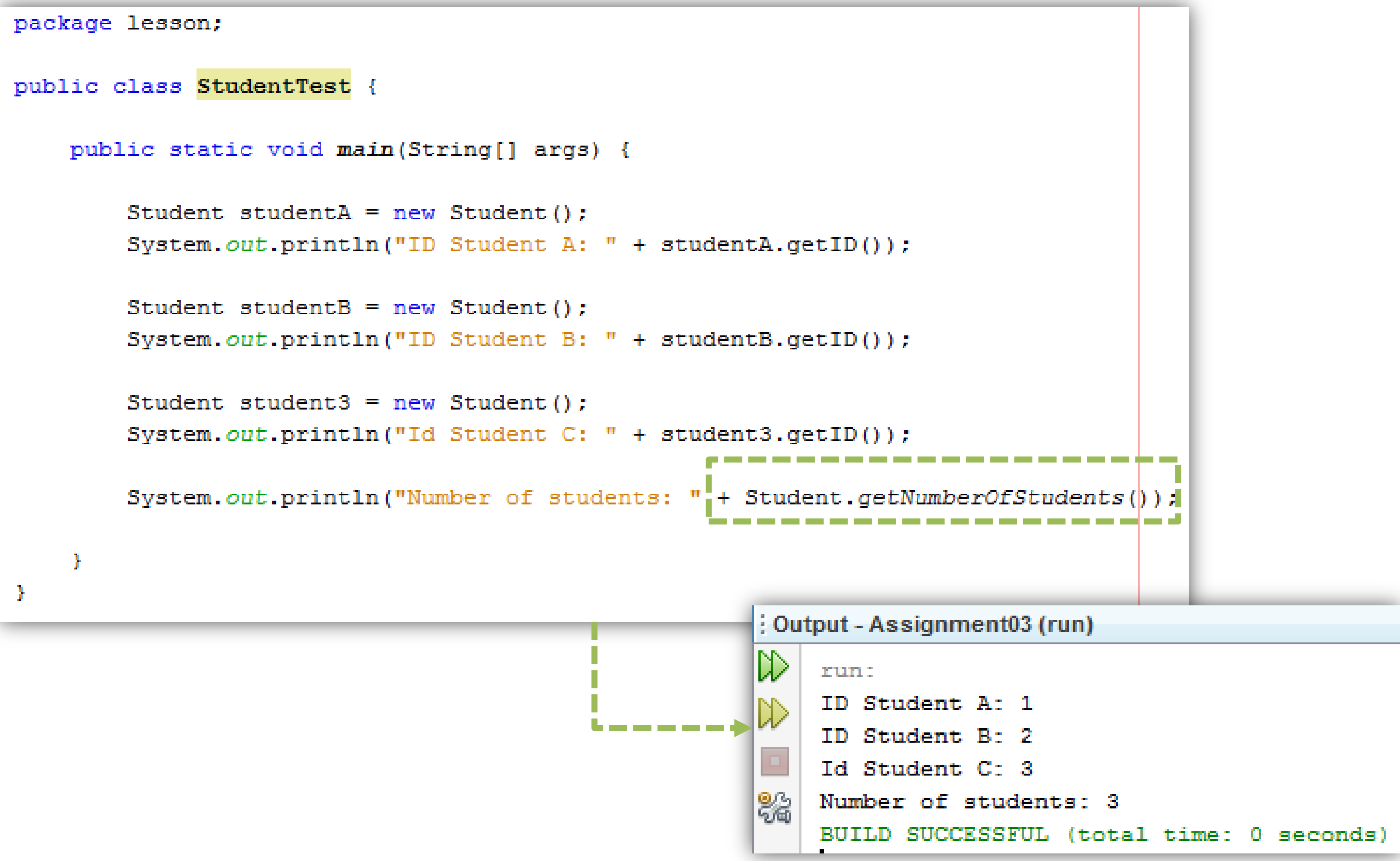
Static attributes are **common to all instanced** objects

Static attributes are **class attributes**



```
package lesson;

public class Student {

    private int id;
    private static int numberOfStudents = 0;

    // ...
    public Student() {
        id = ++numberOfStudents;
    }

    public int getID() {
        return id;
    }

    public static int getNumberOfStudents() {
        return numberOfStudents;
    }
    // ...
}
```

Static attributes example



```java
package lesson;

public class StudentTest {

    public static void main(String[] args) {

        Student studentA = new Student();
        System.out.println("ID Student A: " + studentA.getID());

        Student studentB = new Student();
        System.out.println("ID Student B: " + studentB.getID());

        Student student3 = new Student();
        System.out.println("Id Student C: " + student3.getID());

        System.out.println("Number of students: " + Student.getNumberOfStudents());

    }
}
```

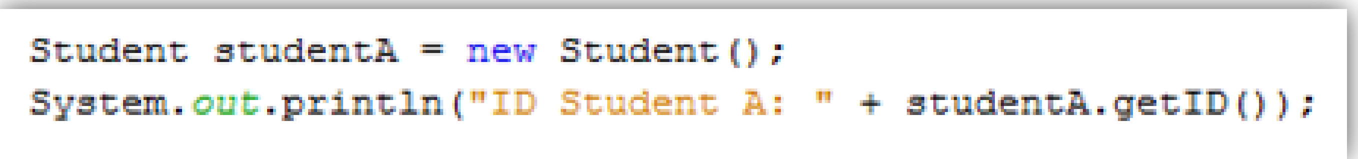
Output - Assignment03 (run)

```
run:
ID Student A: 1
ID Student B: 2
Id Student C: 3
Number of students: 3
BUILD SUCCESSFUL (total time: 0 seconds)
```

Static attributes example

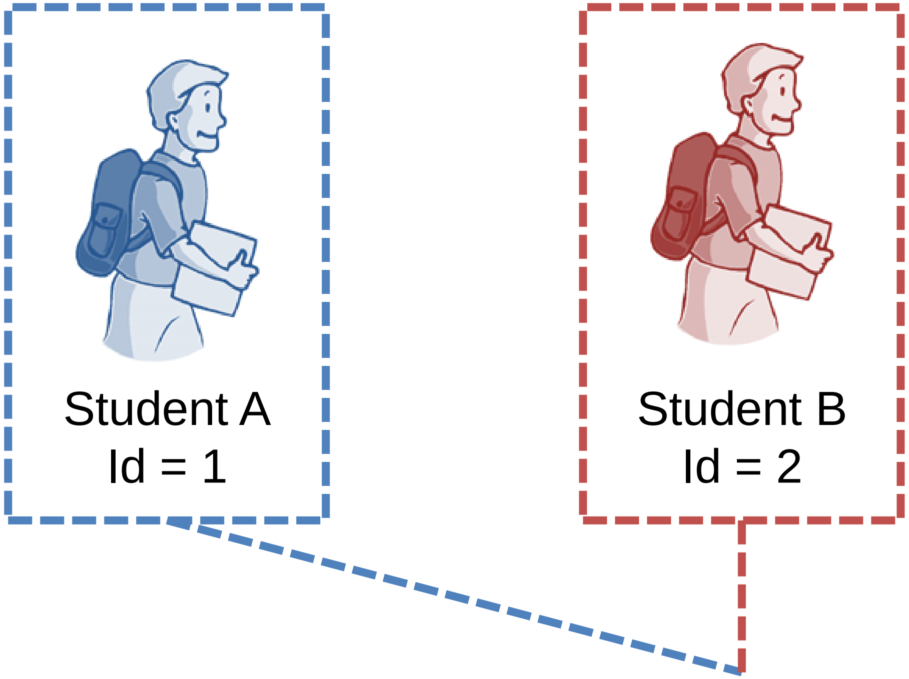
Student A
Id = 1

Static property Number of students = **1**



```
Student studentA = new Student();
System.out.println("ID Student A: " + studentA.getID());
```

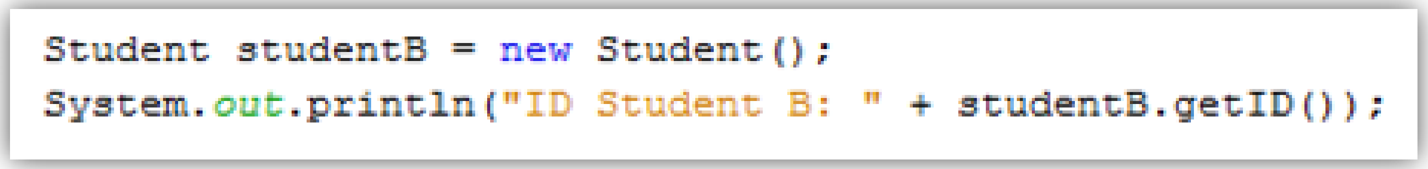
Static attributes example



Student A
Id = 1

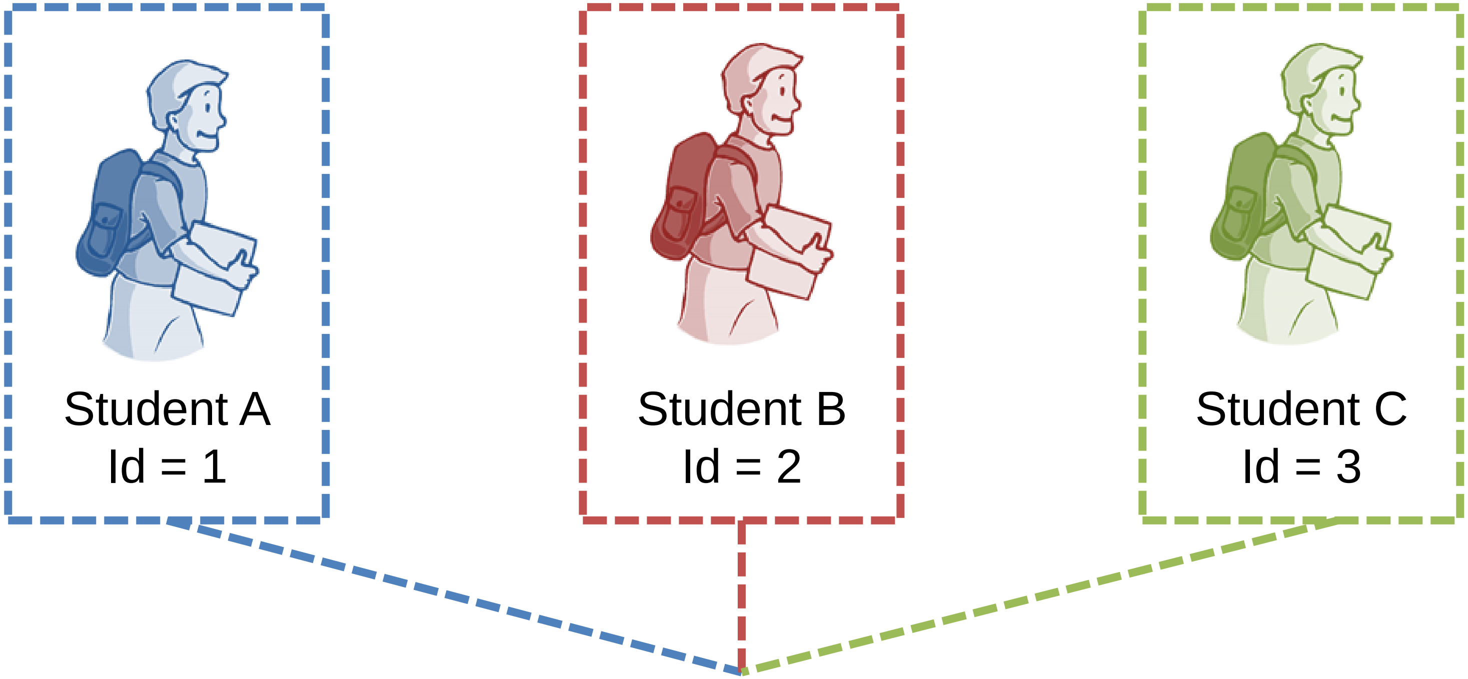
Student B
Id = 2

Static property Number of students = **2**



```
Student studentB = new Student();
System.out.println("ID Student B: " + studentB.getID());
```

Static attributes example

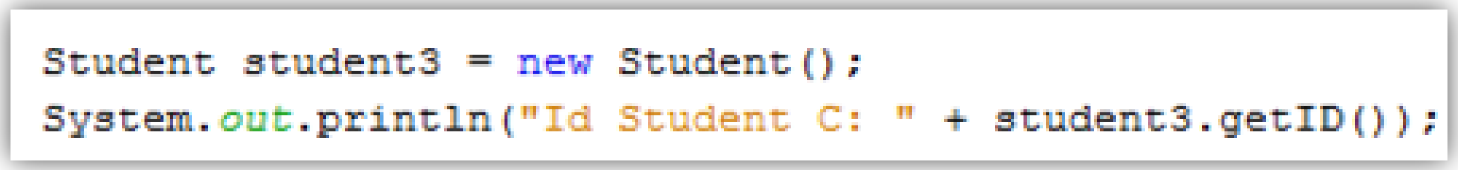


Student A
Id = 1

Student B
Id = 2

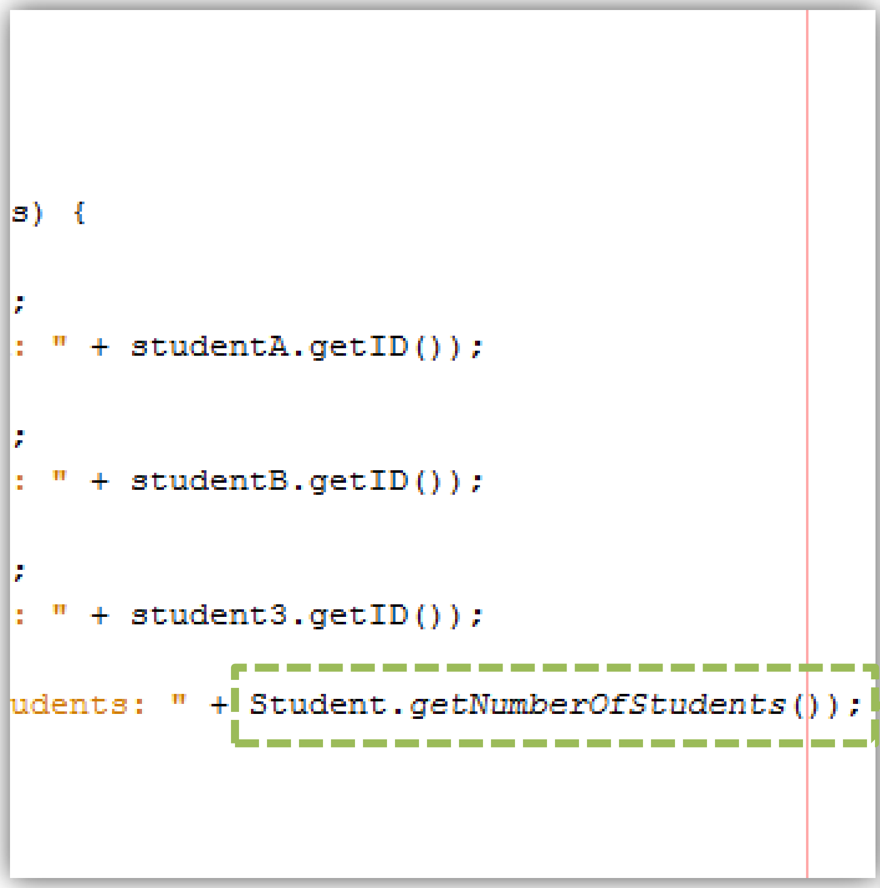
Student C
Id = 3

Static property Number of students = **3**

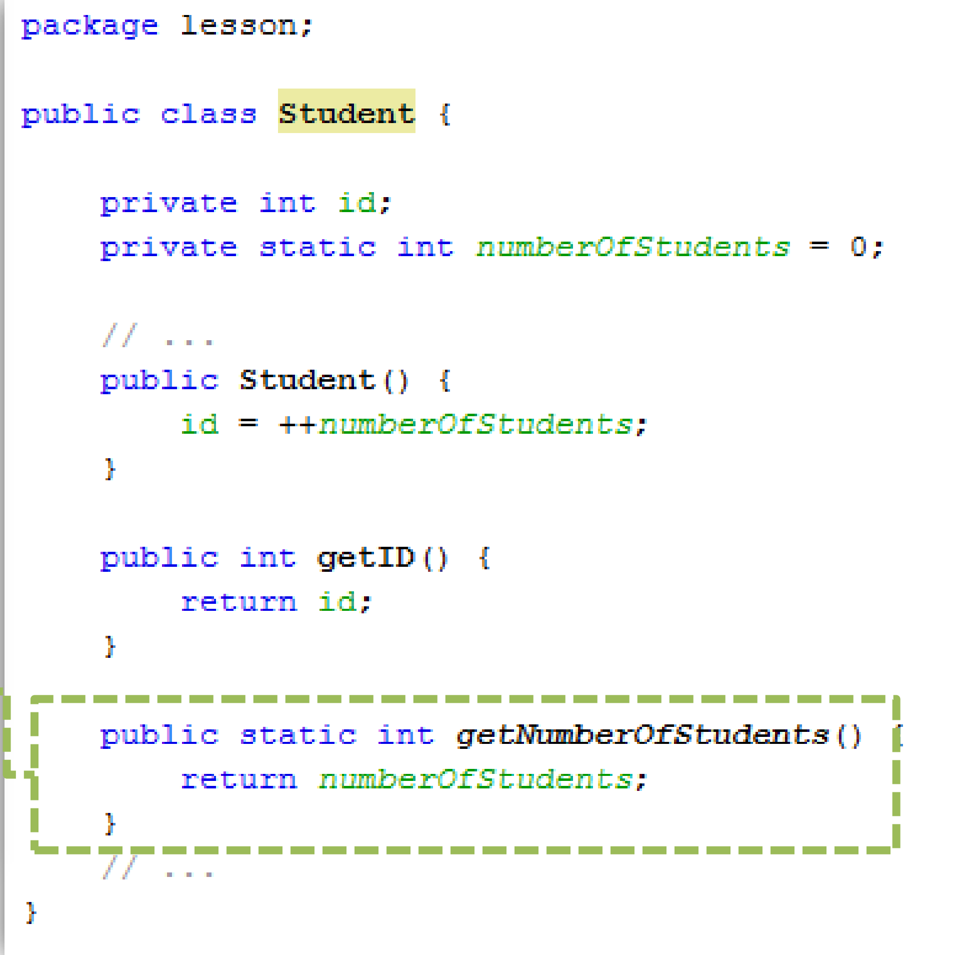


```
Student student3 = new Student();
System.out.println("Id Student C: " + student3.getID());
```

Accessing `static` attributes



```
s) {

;
: " + studentA.getID());

;
: " + studentB.getID());

;
: " + student3.getID());

udents: " + Student.getNumberOfStudents());
```

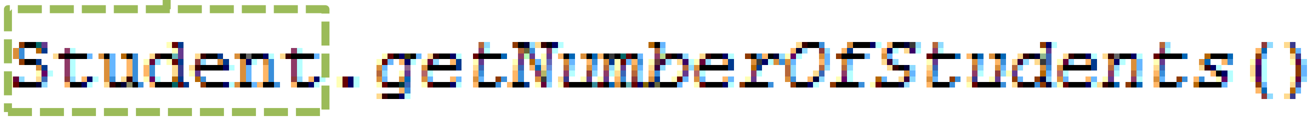


```
package lesson;

public class Student {

    private int id;
    private static int numberOfStudents = 0;

    // ...
    public Student() {
        id = ++numberOfStudents;
    }

    public int getID() {
        return id;
    }

    public static int getNumberOfStudents() {
        return numberOfStudents;
    }
    // ...
}
```

1.2 Static Methods

Accessing `static` methods or attributes

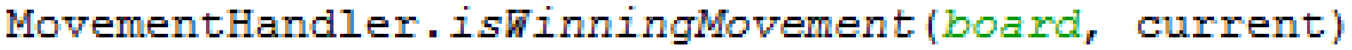
We **do not need to instantiate any object** to access static attributes or methods



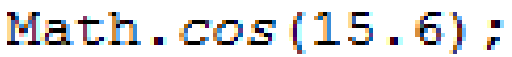
```
Student.getNumberOfStudents()
```

Examples

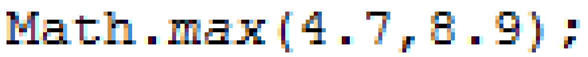
Static methods typically take all they data from parameters and compute something from those parameters.



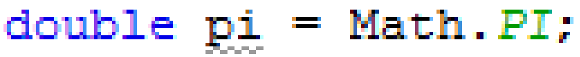
```
MovementHandler.isWinningMovement(board, current)
```



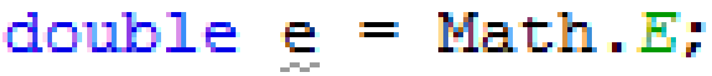
```
Math.cos(15.6);
```



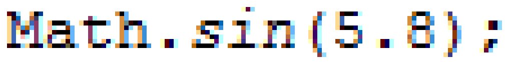
```
Math.max(4.7,8.9);
```



```
double pi = Math.PI;
```

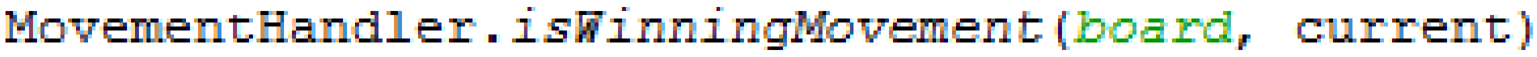


```
double e = Math.E;
```

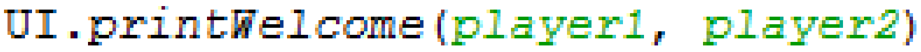


```
Math.sin(5.8);
```

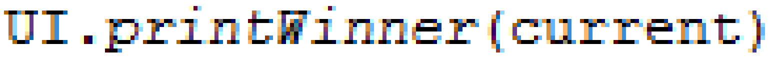
TicTacToe Examples



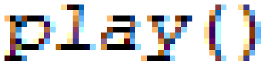
```
MovementHandler.isWinningMovement(board, current)
```



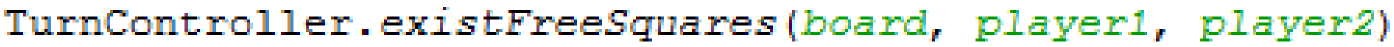
```
UI.printWelcome(player1, player2)
```



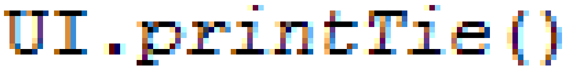
```
UI.printWinner(current)
```



```
play()
```



```
TurnController.existFreeSquares(board, player1, player2)
```

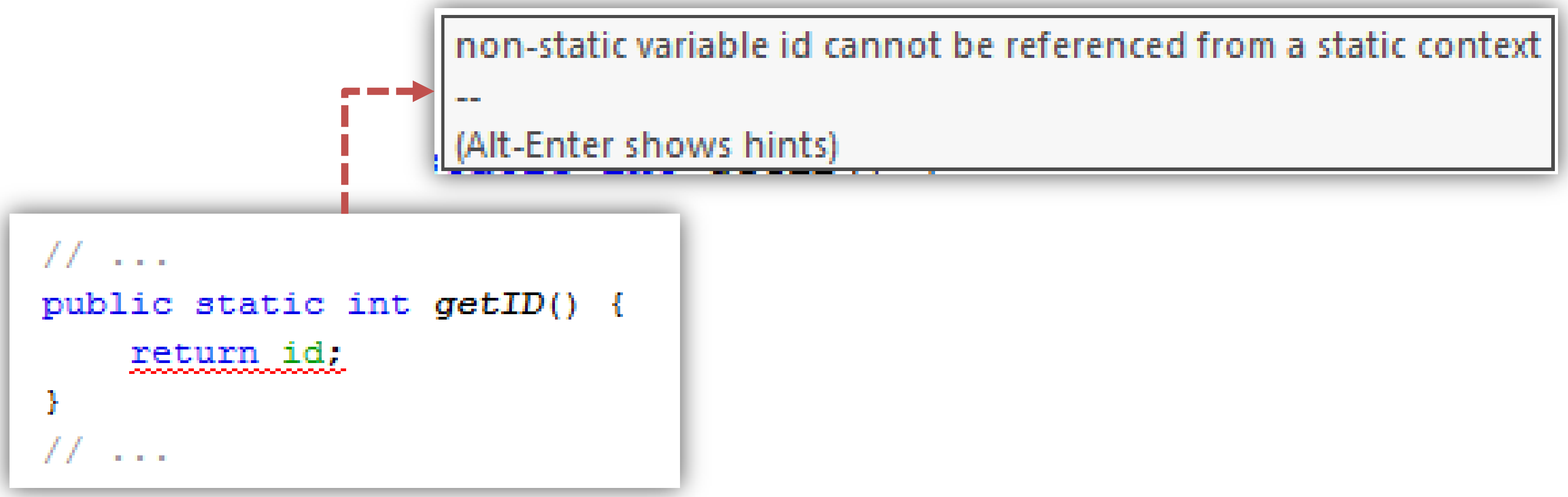


```
UI.printTie()
```

Accessing `static` methods or attributes

Be careful!!!

Static methods cannot access Non-Static attributes



```
non-static variable id cannot be referenced from a static context
--
(Alt-Enter shows hints)
```

```java
// ...
public static int getID() {
    return id;
}
// ...
```

References

[Barker] J. Barker, *Beginning Java Objects: From Concepts To Code*, Second Edition, Apress, 2005.

[Oracle] *Understanding Instance and Class Members,* Available: http://download.oracle.com/javase/tutorial/java/javaOO/classvars.html

[Oracle] Java API documentation, *Class Object,* Available: http://download.oracle.com/javase/6/docs/api/java/lang/Object.html