EECS 1001 and EECS 1030M, lab 01 conflict

Those students who are taking EECS 1001 and who are enrolled in lab 01 of EECS 1030M should switch to lab 02. If you need my help with switching lab sections, please let me know.

Office hours

- In LAS 3046: Mondays, 16:30-17:20 and Thursdays, 13:30-14:20
- In the lab: Wednesdays, 16:00-17:20, Thursdays, 14:30-15:50 (in the weeks there are no tests scheduled)
- By appointment (send me email)

Test 1

- When: next week, during your lab
- What: Chapter 2, excluding Section 2.6
- Type of questions: similar to the questions of the Quiz on Monday, plus one more challenging question
- Textbook: study it, since studying just the slides might not be enough
- Lab: attend the lab in which you are officially enrolled so that we can ensure that there is a computer for everyone

Chapter 3: Implementing Non-Static Features

EECS 1030

moodle.yorku.ca

Implementing non-static features

Problem

Implement the Rectangle class.

Phase 1: analysis

Study the API carefully.

Phase 1: analysis

Study the API carefully.

Question

The methods `equals` and `hashCode` and `toString` are called obligatory methods. Why?

Phase 1: analysis

Study the API carefully.

Question

The methods `equals` and `hashCode` and `toString` are called obligatory methods. Why?

Answer

Because the `Object` class contains these methods and, therefore, each class has them (either inherits them or overrides them).

Let's be a client

Write an app that

- creates a `Rectangle` with width 1 and height 2,
- prints the rectangle,
- scales the rectangle by a factor 2, and
- prints the rectangle again.

Memory diagram

Draw the memory diagram representing memory after the rectangle has been printed for the first time.

Invocation of a static method

Question

When executing `Math.sqrt(2)`, how many pieces of data are passed to the method invocation?

Invocation of a static method

Question

When executing `Math.sqrt(2)`, how many pieces of data are passed to the method invocation?

Answer

One, namely the value 2.

Invocation of a static method

Question

When executing `Math.sqrt(2)`, how many pieces of data are passed to the method invocation?

Answer

One, namely the value 2.

Question

Draw the invocation block for `Math.sqrt(2)`.

Invocation of a non-static method

Question

When executing `rectangle.scale(2)`, how many pieces of data are passed to the method invocation?

Invocation of a non-static method

Question

When executing `rectangle.scale(2)`, how many pieces of data are passed to the method invocation?

Answer

Two, namely the value 2 and the value of the object reference `rectangle`.

this

When executing `rectangle.scale(2)`, two arguments are passed to the method invocation.

The `scale` method has only one (explicit) parameter, called `factor`. The other parameter is implicit and is called `this`.

this

When executing `rectangle.scale(2)`, two arguments are passed to the method invocation.

The `scale` method has only one (explicit) parameter, called `factor`. The other parameter is implicit and is called `this`.

Question

Draw the invocation block for `rectangle.scale(2)`.

Question

What is `this`?

this

Question

What is `this`?

Answer

A Java keyword and an implicit parameter of non-static methods and constructors.

this

Question

What is `this`?

Answer

A Java keyword and an implicit parameter of non-static methods and constructors.

Question

What does `this` capture?

this

Question

What is `this`?

Answer

A Java keyword and an implicit parameter of non-static methods and constructors.

Question

What does `this` capture?

Answer

A reference to the object on which the method/constructor is invoked.

Structure of a class

```
\\ package statement

\\ import statements

public class ClassName
{
   \\ attributes

   \\ constructors

   \\ methods
}
```

Attributes

Question

Does the API of the `Rectangle` class contain any public attributes?

Attributes

Question

Does the API of the `Rectangle` class contain any public attributes?

Answer

No.

Attributes

Question

Does the API of the `Rectangle` class contain any public attributes?

Answer

No.

Question

Should the `Rectangle` class contain any attributes?

Attributes

Question

Does the API of the `Rectangle` class contain any public attributes?

Answer

No.

Question

Should the `Rectangle` class contain any attributes?

Answer

Yes, because the state of a rectangle should contain the width and height of the rectangle.

All attributes of the `Rectangle` class are private.

Attributes

Question

Which private attributes do we introduce to capture the state of a rectangle? Provide both its type and a descriptive name.

Attributes

Question

Which private attributes do we introduce to capture the state of a rectangle? Provide both its type and a descriptive name.

Answer

width : int
height : int

Attributes

Question

Could the width and height of a rectangle be represented differently?

Attributes

Question

Could the width and height of a rectangle be represented differently?

Answer

Yes. The state of a rectangle with width 1 and height 2 can be represented, for example, as the `String` "1-2" and as the `long` $4294967298 = 1 \times 2^{32} + 2$.

Initialize the state

Question

Where do we initialize the state of an object?

Initialize the state

Question

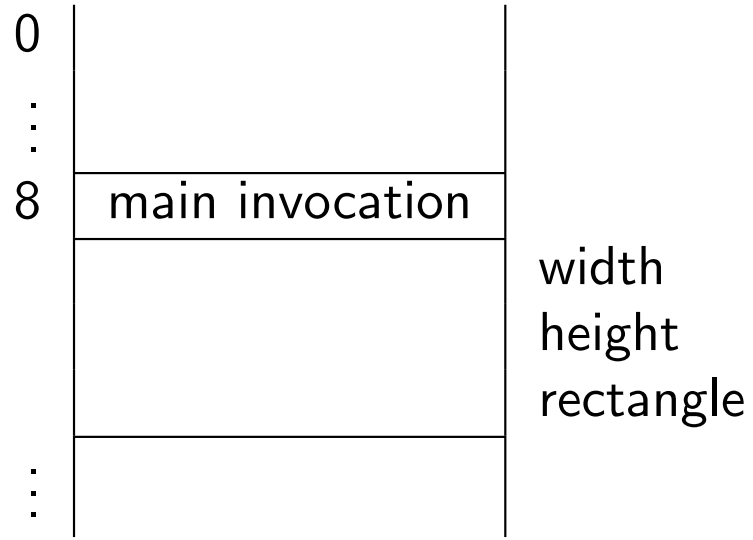
Where do we initialize the state of an object?

Answer

In the constructor.

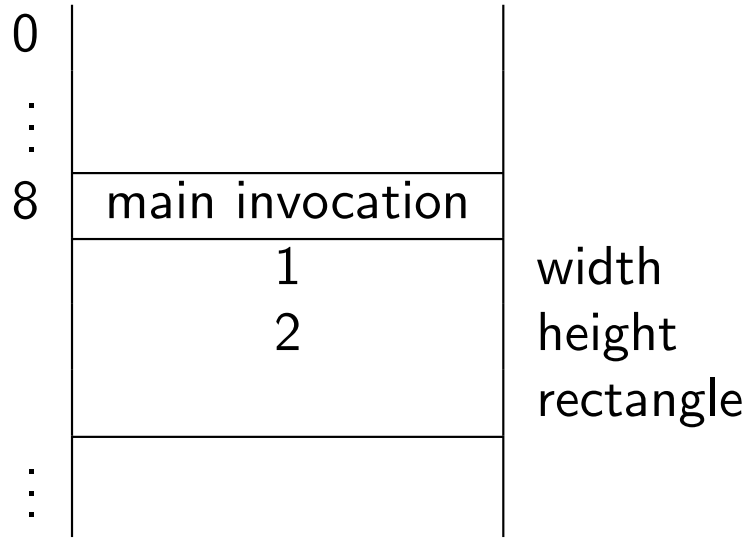
Memory diagrams

```
int width = 1;
int height = 2;
Rectangle rectangle = new Rectangle(width, height);
```



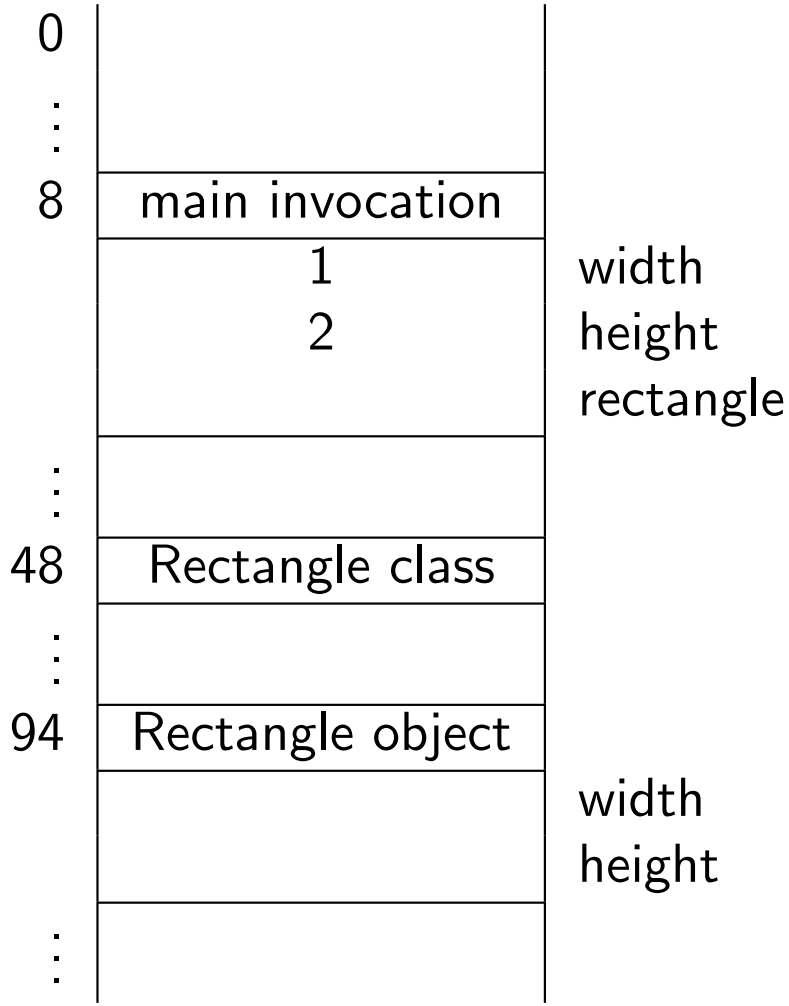
Memory diagrams

```
int width = 1;
int height = 2;
Rectangle rectangle = new Rectangle(width, height);
```



Memory diagrams

```
int width = 1;
int height = 2;
Rectangle rectangle = new Rectangle(width, height);
```



Memory diagrams

```
int width = 1;
int height = 2;
Rectangle rectangle = new Rectangle(width, height);
```