Programming for Mobile Computing

EECS 1022

`moodle.yorku.ca`

Midterm: written part

When: Monday July 10, 18:30–19:30

What: material covered in Lecture 1–5

Note: 40 minute lecture after midterm

Midterm: written part

- No questions are allowed during the test. If a question is not clear, then write down any assumptions made.
- One page of notes (letter size, double sided) may be used during the test.
- A non-electronic dictionary may be used during the test.

Midterm: written part

- Answer each question in the space provided.
- Make sure that you have answered all questions (test is double sided).
- Manage your time carefully.
- Last page can be used as scrap paper.

Midterm: written part

Preparation

- Study the material.
- Prepare your page of notes.
- Think of a test question.
- Post your question on the forum at Moodle.
- Answer questions posted by other students on the forum.
- Discuss questions and answers on the forum.

Midterm: programming part

When: Tuesday July 11, during your lab

What: material covered in Lab 1, 3 and 5

Midterm: programming part

- Access will be provided to the lecture slides and the sample code.
- Access will be provided to the Java Standard Library API and the Android API.
- Most likely no access to the Internet will be provided.
- WSC laptops needs to be used. You cannot use your own laptop.
- Tablets cannot be used during the test.

Midterm: programming part

What you can use during the programming test:

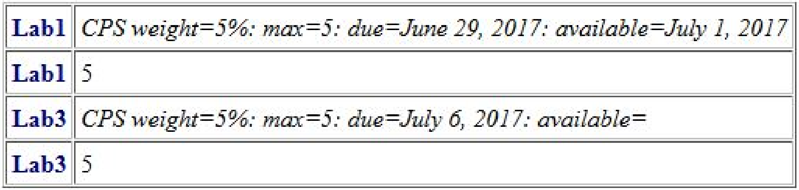
- WSC laptop.
- A non-electronic dictionary.
- One piece of paper (will be provided).
- Pen or pencil.
- Student card.

Academic honesty

If a student behaves academically dishonest during the test, this will be reported to Lassonde's Assistant Dean. If found guilty of academic dishonesty during the meeting with the Assistant Dean, then I will suggest a zero for the test as penalty.

Lab marks

Check if your mark for Lab 1 and 3 has been recorded at the URL https://www.eecs.yorku.ca/~roumani/ePost/ppy/ep.cgi?year=2016-17&term=S&course=1022 (log in with your Passport York credentials). You should see something like the following screenshot.



Lab1	*CPS weight=5%: max=5: due=June 29, 2017: available=July 1, 2017*
Lab1	5
Lab3	*CPS weight=5%: max=5: due=July 6, 2017: available=*
Lab3	5

If you cannot find your marks, please show Lab 1 and 3 to the teaching assistant on Thursday July 6 during the lab (this is the last opportunity).

API of the Rectangle class

The API of the Rectangle class can be found at the URL `www.eecs.yorku.ca/course_archive/2016-17/S/1022/api/rectangle.api/`

The Rectangle class

Question

1. How many attributes should we introduce?
2. What are their types?
3. What are appropriate descriptive names?

The `Rectangle` class

Question

1. How many attributes should we introduce?
2. What are their types?
3. What are appropriate descriptive names?

Answer

1. Two.
2. `int`.
3. `width` and `height`.

Creation of an object

Question

Create a $\mathrm{Rectangle}$ with width 1 and height 2 named $\mathrm{rectangle}$.

Creation of an object

Question

Create a Rectangle with width 1 and height 2 named rectangle.

Answer

Rectangle rectangle = new Rectangle(1, 2);

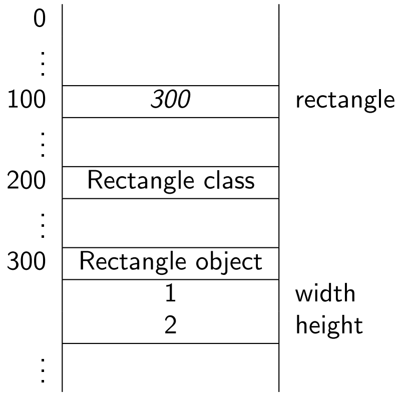
Memory diagram

Draw the memory diagram representing memory after execution of

Rectangle rectangle = new Rectangle(1, 2);

Memory diagram

```
Rectangle rectangle = new Rectangle(1, 2);
```



Invocation of a method

Question

Make rectangle twice as large using the scale method.

Invocation of a method

Question

Make rectangle twice as large using the scale method.

Question

```
Rectangle rectangle = new Rectangle(1, 2);
rectangle.scale(2);
```

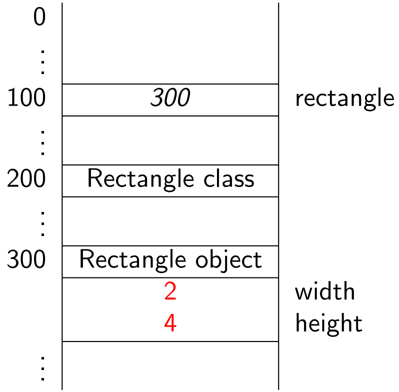
Memory diagram

Draw the memory diagram representing memory after execution of

```
Rectangle rectangle = new Rectangle(1, 2);
rectangle.scale(2);
```

Memory diagram

```
Rectangle rectangle = new Rectangle(1, 2);
rectangle.scale(2);
```



Invocation of a method

Question

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

Invocation of a 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`.

Invocation of a 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`.

Question

How many explicit parameters does the `scale` method have?

Invocation of a 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`.

Question

How many explicit parameters does the `scale` method have?

Answer

One.

`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`

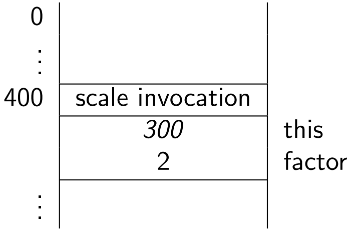
When executing `rectangle.scale(2)`, two arguments are passed to the method invocation.

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)`.

Memory diagram



`this`

Question

What is `this`?

`this`

Question

What is `this`?

Answer

A Java keyword and an implicit parameter of methods and constructors.

`this`

Question

What is `this`?

Answer

A Java keyword and an implicit parameter of methods and constructors.

Question

What does `this` capture?

`this`

Question

What is `this`?

Answer

A Java keyword and an implicit parameter of methods and constructors.

Question

What does `this` capture?

Answer

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

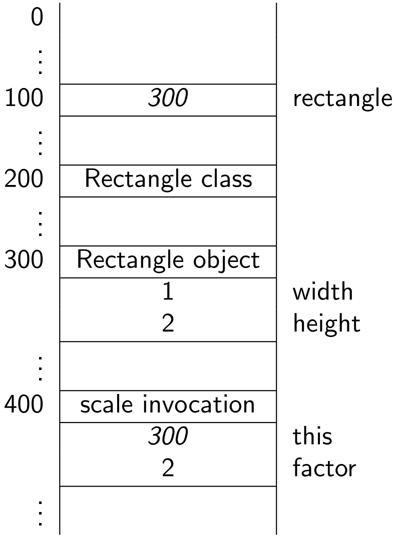
The scale method

Question

Implement the scale method.

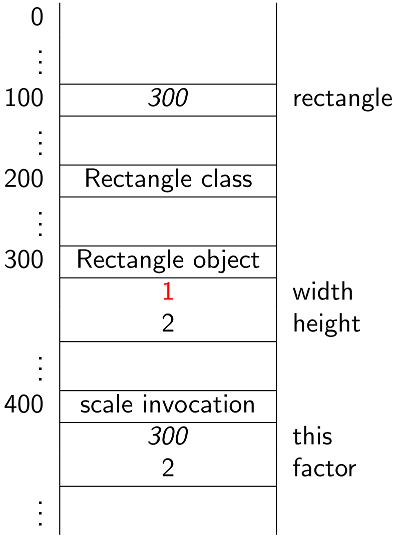
Memory diagram

```
this.width = this.width * factor;
this.height = this.height * factor;
```



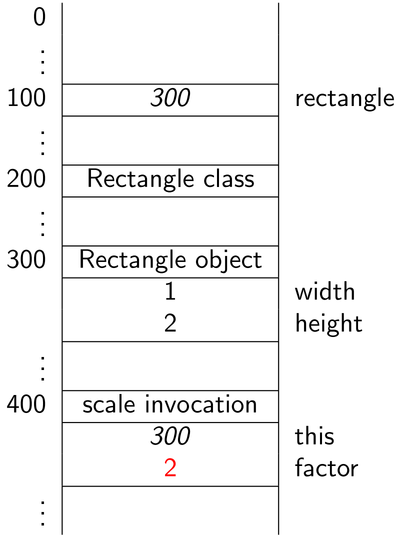
Memory diagram

```
this.width = this.width * factor; // 1 * factor
this.height = this.height * factor;
```



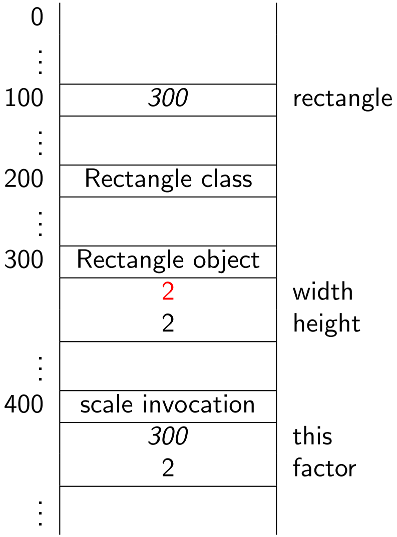
Memory diagram

```
this.width = this.width * factor; // 1 * 2
this.height = this.height * factor;
```



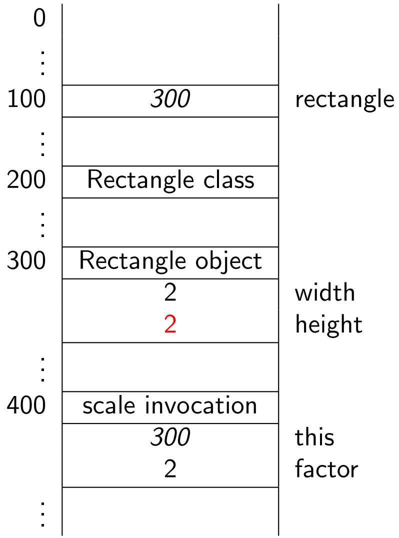
Memory diagram

```
this.width = this.width * factor; // 1 * 2
this.height = this.height * factor;
```



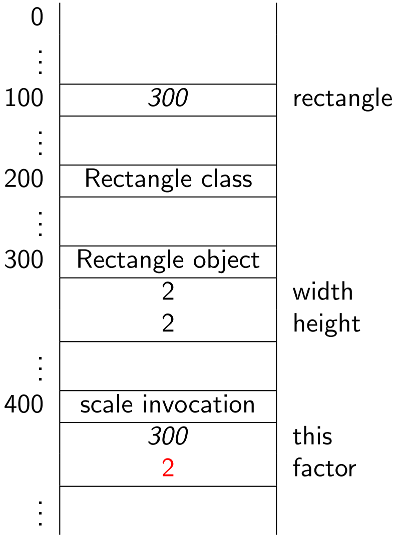
Memory diagram

```
this.width = this.width * factor; // 1 * 2
this.height = this.height * factor; // 2 * factor
```



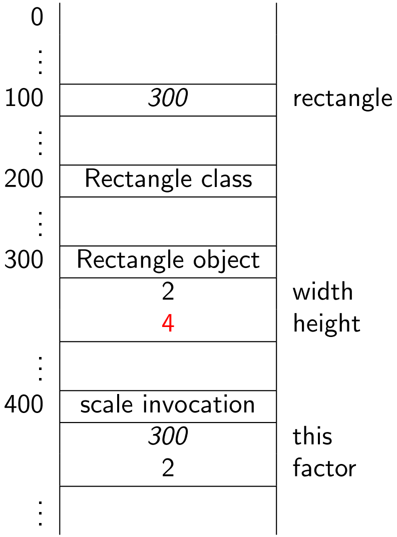
Memory diagram

```
this.width = this.width * factor;  // 1 * 2
this . height  = this.height * factor; // 2 * 2
```



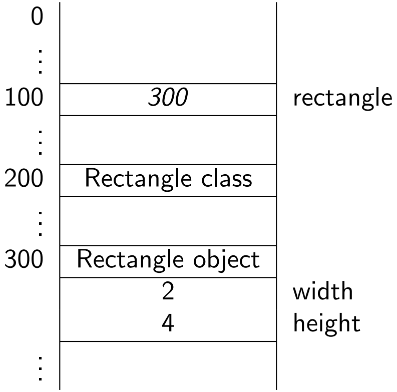
Memory diagram

```
this.width = this.width * factor; // 1 * 2
this.height = this.height * factor; // 2 * 2
```



Memory diagram

```
this.width = this.width * factor; // 1 * 2
this.height = this.height * factor; // 2 * 2
```



Invocation of a method

Question

Store the area of $\mathrm{rectangle}$ in a variable named area.

Invocation of a method

Question

Store the area of rectangle in a variable named area.

Question

```
Rectangle rectangle = new Rectangle(1, 2);
rectangle.scale(2);
int area = rectangle.getArea();
```

Invocation of a method

Question

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

Invocation of a method

Question

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

Answer

One, namely the value of the object reference `rectangle`.

Invocation of a method

Question

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

Answer

One, namely the value of the object reference `rectangle`.

Question

How many explicit parameters does the `getArea` method have?

Invocation of a method

Question

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

Answer

One, namely the value of the object reference `rectangle`.

Question

How many explicit parameters does the `getArea` method have?

Answer

Zero.

`this`

When executing `rectangle.getArea()`, one arguments is passed to the method invocation.

The `getArea` method has no (explicit) parameter. Its implicit parameter is called `this`.

`this`

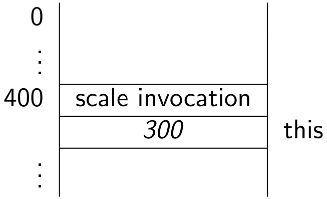
When executing `rectangle.getArea()`, one arguments is passed to the method invocation.

The `getArea` method has no (explicit) parameter. Its implicit parameter is called `this`.

Question

Draw the invocation block for `rectangle.getArea()`.

Memory diagram



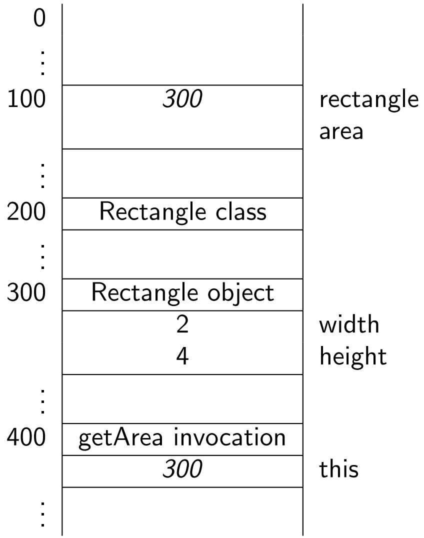
The getArea method

Question

Implement the getArea method.

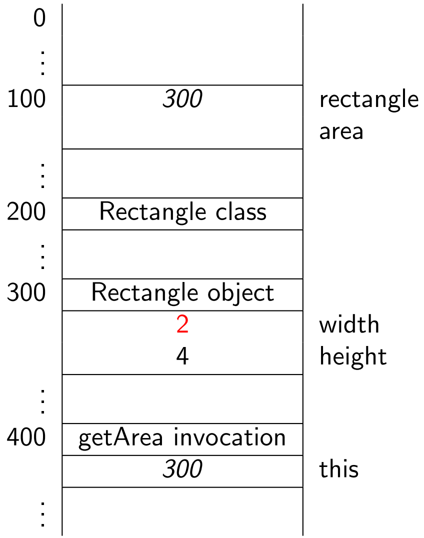
Memory diagram

```
return this.width * this.height;
```



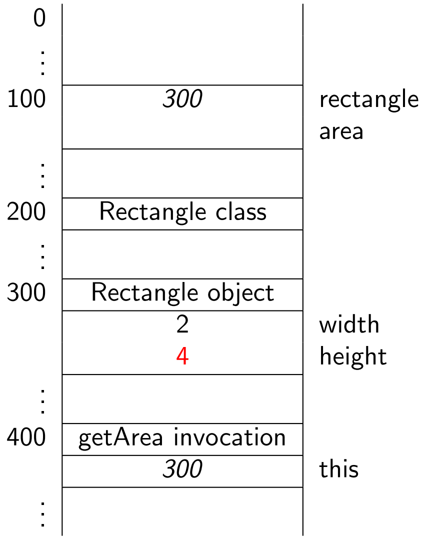
Memory diagram

```
return this.width * this.height; // 2 * this.height
```



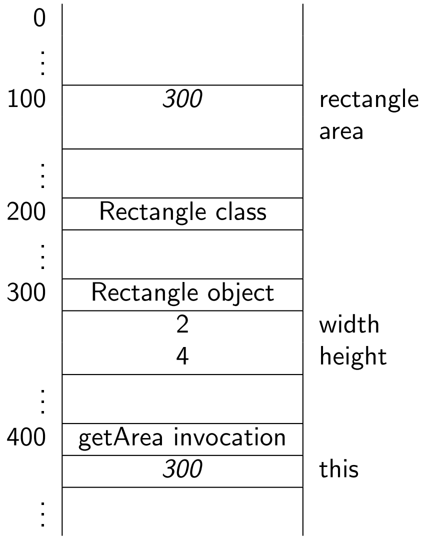
Memory diagram

```
return this.width * this.height; // 2 * 4
```



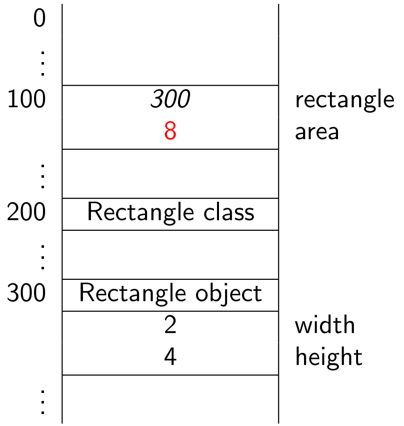
Memory diagram

```
return this.width * this.height; // 8
```



Memory diagram

```
Rectangle rectangle = new Rectangle(1, 2);
rectangle.scale(2);
int area = rectangle.getArea(); // 8
```



Initialize the state

Question

Where do we initialize the state of an object?

Initialize the state

Question

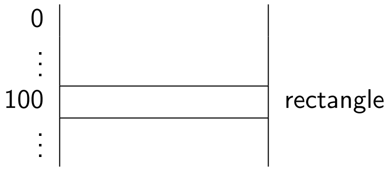
Where do we initialize the state of an object?

Answer

In the constructor.

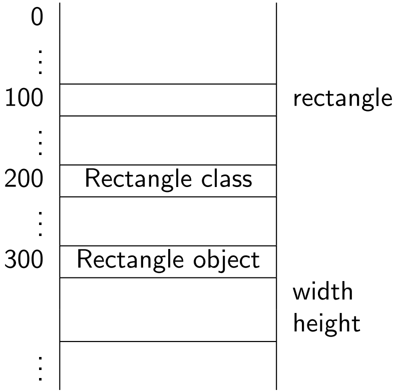
Memory diagrams

```
Rectangle rectangle = new Rectangle(1, 2);
```



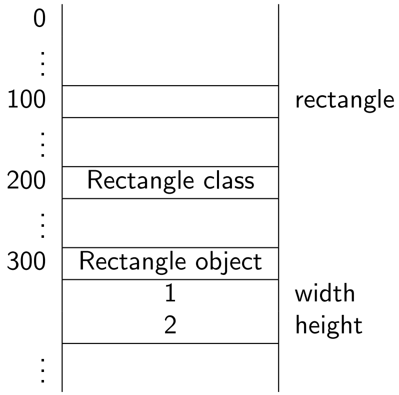
Memory diagrams

```
Rectangle  rectangle  = new Rectangle(1, 2);
```



Memory diagrams

```
Rectangle rectangle = new Rectangle(1, 2);
```



Invocation of a constructor

Question

When executing the constructor `Rectangle(1, 2)`, how many pieces of data are passed to the method invocation?

Invocation of a constructor

Question

When executing the constructor `Rectangle(1, 2)`, how many pieces of data are passed to the method invocation?

Answer

Three, namely the values 1 and 2 and the value of the new object reference.

Invocation of a constructor

Question

When executing the constructor `Rectangle(1, 2)`, how many pieces of data are passed to the method invocation?

Answer

Three, namely the values 1 and 2 and the value of the new object reference.

Question

How many explicit parameters does this constructor have?

Invocation of a constructor

Question

When executing the constructor `Rectangle(1, 2)`, how many pieces of data are passed to the method invocation?

Answer

Three, namely the values 1 and 2 and the value of the new object reference.

Question

How many explicit parameters does this constructor have?

Answer

Two

`this`

When executing `Rectangle(1, 2)`, three arguments are passed to the constructor invocation.

This constructor two (explicit) parameters and implicit parameter called `this`.

`this`

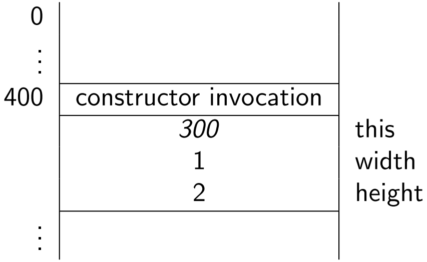
When executing `Rectangle(1, 2)`, three arguments are passed to the constructor invocation.

This constructor two (explicit) parameters and implicit parameter called `this`.

Question

Draw the invocation block for `Rectangle(1, 2)`.

Memory diagram



The constructor

Question

Implement the constructor.