Lab 2 Report

Steven Mayher & Tyler Youk
June 8, 2022

Introduction to Programming in Java: Lab 2
Calling Java Methods (Functions)
Creating Java Objects

In this lab you will learn how to call some of the pre-defined functions in Java as well as how to to manipulate objects by playing with the Math and JFrame classes and entering Java code fragments into the DrJava interactions pane.

If you did not join the recitation, you need to complete the first 6 tasks (including task 6).

Task 1. Assigning Lab Partners

The first thing that will occur in this lab is the assigning of lab partners. The teaching assistants in the recitation will pair you with another student.

Rules For The Lab

Your grade will be determined by how well you work as a member of a team. You will play two different roles in the lab. You will either be the *driver* or *navigator*.

- **driver:** The person typing at the keyboard.
- **navigator:** The person watching for mistakes, making suggestions, and thinking ahead.

Important: The driver is the only one who may do the typing. The navigator must not jump in and start editing.

Choose one of you to begin as the driver. The other will be the navigator, and the lab will tell you when to switch roles.

You do not have to complete all the tasks by the end of lab, but you do have to make a good effort.

The only ways to lose points when working as a team is 1) you do not show up on time, 2) you goof off, 3) you are the navigator and you take over editing the document from the driver, or 4) you sit quietly as the navigator and never offer suggestions.

Lab 2 Report

Steven Mayher & Tyler Youk
June 8, 2022

If you are unable to attend the recitation then you must do this lab exercise on your own. In that case, your grade will be the percent of the exercises you complete.

Submitting the Lab

The form of this lab is similar to the one from last week. You are given a series of tasks to do and then some questions to answer. Answer the questions in a lab report and submit the report at the end of lab. *Both lab partners should submit the lab report.*

If you are new to programming, your goal is to make it at least through Task 5. If you have not completed step 5 and we are getting close to the end of the lab, ask the lab teaching assistants for help.

Task 2. Introductions

Tell your lab partner your name as well as a sport or game you love to play.

Choose one of you to begin as the driver. The other will be the navigator, and the lab will tell you when to switch roles.

Task 3. Starting DrJava and Starting a Lab Report

Launch DrJava on the lab computer. Just as you did in the last lab, you will create a lab report. I recommend creating a shared Googld document. Make sure to put both lab partners' names at the top of the report.

Task 4. Playing with JFrames.

We will begin the lab with the pre-defined class JFrame. The JFrame class represents a window on the screen. To use a pre-defined class, we must first load it. The JFrame class is in the javax.swing package, and so the the Java statement to load a class is

```
import javax.swing.JFrame;
```

Lab 2 Report

Steven Mayher & Tyler Youk
June 8, 2022

Type that into the DrJava interactions pane now.

Now that we have the JFrame class loaded, we can use it.

Answers in Maroon

1. Enter the following lines, and in your lab report, describe what happens after each statement.

import javax.swing.JFrame; **= no output in interactions pane (imports JFrame class)**
JFrame j1 = new JFrame(); **= no output in interactions pane (declares j1 as a new JFrame)**
j1.setVisible(true); **= produces a tiny java window with just the navigation buttons**
j1.setSize(200,400); **= changes the size of the java window to the width & height specified (set Size to (200,400))**
j1.setLocation(500,500); **= no error / output in interactions pane (set Location to (500,500))**

Now try creating a few more JFrame variables. You can call the variables (almost) anything you wish. Try calling the following methods:

setVisible	getWidth
setSize	getHeight
setLocation	getX
	getY

setVisible expects a single input of type boolean, setSize and setLocation each expect two values of type int (remember to separate the values by a comma), and getWidth, getHeight, getX and getY do not expect any input.

Also try calling each method with a semicolon after them and without a semicolon after them. If you place a semicolon at the end of a line, it is a Java *statement*. A statement does not have a value. If you do not place a semicolon at the end, it is a Java *expression*, and expressions may return values. By omitting the semicolon, you will see any values that an expression may return.

(with no semicolon)
setVisible(boolean) = no error, sets JFrame to visible
setSize(var, var) = no error, sets Size
setLocation(var, var) = no error, sets Location
getWidth() = no error, outputs width
getHeight() = no error, outputs height
getX() = no error, outputs x value of location
getY() = no error, outputs y value of location

(with semicolon)
setVisible(boolean); = no error, sets JFrame to visible
setSize(var, var); = no error, sets Size
getWidth(); = doesn't produce output
getHeight(); = doesn't produce output

Steven Mayher & Tyler Youk
June 8, 2022

setLocation(var, var); = no error, sets Location getX(); = doesn't produce output
getY(); = doesn't produce output

2. Answer the following questions in your lab report:

a. What are the *syntactic* differences between the methods in the left column and the methods in the right column? *Syntax* has to do with the actual structure of the line of code: how does the line appear when you type it in? Make a brief list of the syntactic differences you observe.

   The get methods produce another line of output. The set methods do not produce a line of output. Additionally, get methods do not work with a semicolon.

2. Answer the following questions in your lab report:

b. What are the *semantic* differences between the methods in the left column and the methods in the right column? *Semantics* has to do with the meaning of the code: what do the methods do to a JFrame? Make a brief list of the semantic differences you observe.

   The get methods output requested data points of a variable based on the inputs.

   The set methods change the variables of the JFrame object.

Task 5. Creating, Resizing and Moving JFrames

Switch roles. The driver should now navigate, and the navigator should now drive.

Now let us see how much you remember from above.

- Reset the Interactions pane. This erases everything you have done including all variables.
- Create and initialize two JFrame variables, and show (i.e. setVisible) them.
- Use the setSize method to make window 1 300x200.
- Use getWidth and getHeight to get the width and height of window 1 and store the results in two variables called w and h. (What type should the variables be?) *Do this without explicitly typing in the numbers 300 and 200.*
- Use setSize to make window 2 the same dimensions as window 1. *Do this without explicitly typing in the numbers 300 and 200*. Instead use the variables you created above.

3. Enter the statements you used to do the above operations in your lab report.

```
> import javax.swing.JFrame;

> JFrame j1 = new JFrame();

> JFrame j2 = new JFrame();

> j1.setVisible(true);

> j2.setVisible(true);

> j1.setSize(300,200);

> int w

> w = j1.getWidth();

> int h

> h = j1.getHeight();

> w

300

> h

200

> j2.setSize(w,h);
```

Now, use the setLocation method to move the windows so window 1 is flush against the top edge of the screen, halfway across, and window 2 is flush against the bottom edge of the screen, halfway across. (If the Dock (i.e. taskbar/menu of icons) is on the top or bottom of your computer, move the windows agains the right and left edge instead, halfway up.)

4. Enter the two setLocation statements you used in your lab report.

```
> j1.setLocation(2560/2,0);

> j2.setLocation(0,1440/2);
```

Show your answers to Tasks 4 and 5 to your lab teaching assistant.

Lab 2 Report

Steven Mayher & Tyler Youk
June 8, 2022

Task 6. Calling Functions From the Math Class

Switch roles. The driver should now navigate and the navigator should now drive.

The Math class is a pre-defined class that contains common mathematical functions. The Math class is in the java.lang package. All classes in java.lang are automatically loaded in every Java program. This means that we do not need to import the Math class in order to use it.

The methods of the Math class have another interesting property. We do not have to create an instance of the Math class to use the methods. This makes sense because, while the height of a JFrame "belongs" to a specific JFrame, the square root function is universal. Such methods that do not belong to an instance of a class are called *static* or *class* methods.

So, to compute 4 raised to the 5th power, we type in

```
Math.pow(4.0, 5.0)
```

Try that now. Notice that we still need to use the dot, but instead of prefixing the method name with an object reference, we prefix it with the class name. Try typing in JFrame.getWidth() and see what happens.

Another method in Math is the square root method, sqrt that takes a single double as input. For example, Math.sqrt(2.0) returns the square root of 2. Note that we can nest method calls such as Math.sqrt(Math.pow(4.0, 2.0)) first computes the square of 4, and then takes the square root of the result, hopefully returning 4.0. You can nest method calls all you want. All that matters is that the *type* of the value returned by the inside method matches that type expected as input from the outside method. **Always keep track of your types!**

5. Answer the following questions and type your answers in the lab report.

a. Try computing the square root of 2^2, and then computing the square of the square root of 2. Mathematically these are the same, but does Java give you the same answer? Explain why or why not.

These did not give the same answer.

We wrote **Math.sqrt(Math.pow(2.0,2.0))** to compute the square root of 2^2. The output showed to be 2.0. We wrote **Math.pow(Math.sqrt(2.0), 2.0)** to compute the square of the square root of 2. The output showed to be 2.000000000004. The discrepancy is due to rounding issues.

b. Using the fact that $\sin^2(x) + \cos^2(x) = 1$, and using the pow and sqrt methods of Math, create an expression that computes that cosine of an angle that has the sine 0.5. (Do not use variables. Do it in one large expression of nested methods.) When you have an expression that works, add it to your lab report.

Steven Mayher & Tyler Youk
June 8, 2022

(extra work) x is the angle

sqrt(1 - cos^2x) = sin(x)

Cosine of an angle that has the sine .5

Command line:

Math.sqrt(1-Math.pow((Math.sin(0.5)),2.0))

output : .877

Task 7. A Final Challenge

Do the following:

a. Create two JFrame windows and make both visible.
b. Make one window a rectangle, but not a square.
c. Using the setSize, getWidth, and getHeight methods of JFrame and the sqrt method of Math, make the second window a square *such that it has the same area as the first window*. Use the getHeight and getWidth methods and the necessary arithmetic operators to verify your work. *Do this without using any variables except for the two JFrame variables.*

6. Enter into your lab report the Java line or lines of code you used to accomplish this task.

Show your results to your lab teaching assistant.

Submit the Lab

Canvas will allow you to submit your lab report. Please do that now. *Both lab partners should submit the lab report.*

You are now done with the lab! Take a well deserved break.