

Mastery Grids (Java Learning System)

User manual

Contact: Jordan Barria-Pineda
✉️ jab464@pitt.edu



My Progress



Mastery Grids is an interface that gives you access to several types of learning content organized in different topics. They are shown here as grid cells.

Me

This row represents your progress in the topics of the course. Each topic is a cell. Gray means 0% of progress and darker color means more progress.

Show detailed estimations of your knowledge in Java concepts ▾



0% 100%

Points per Topic

means that you got 2 points for completing at least 2 problems.

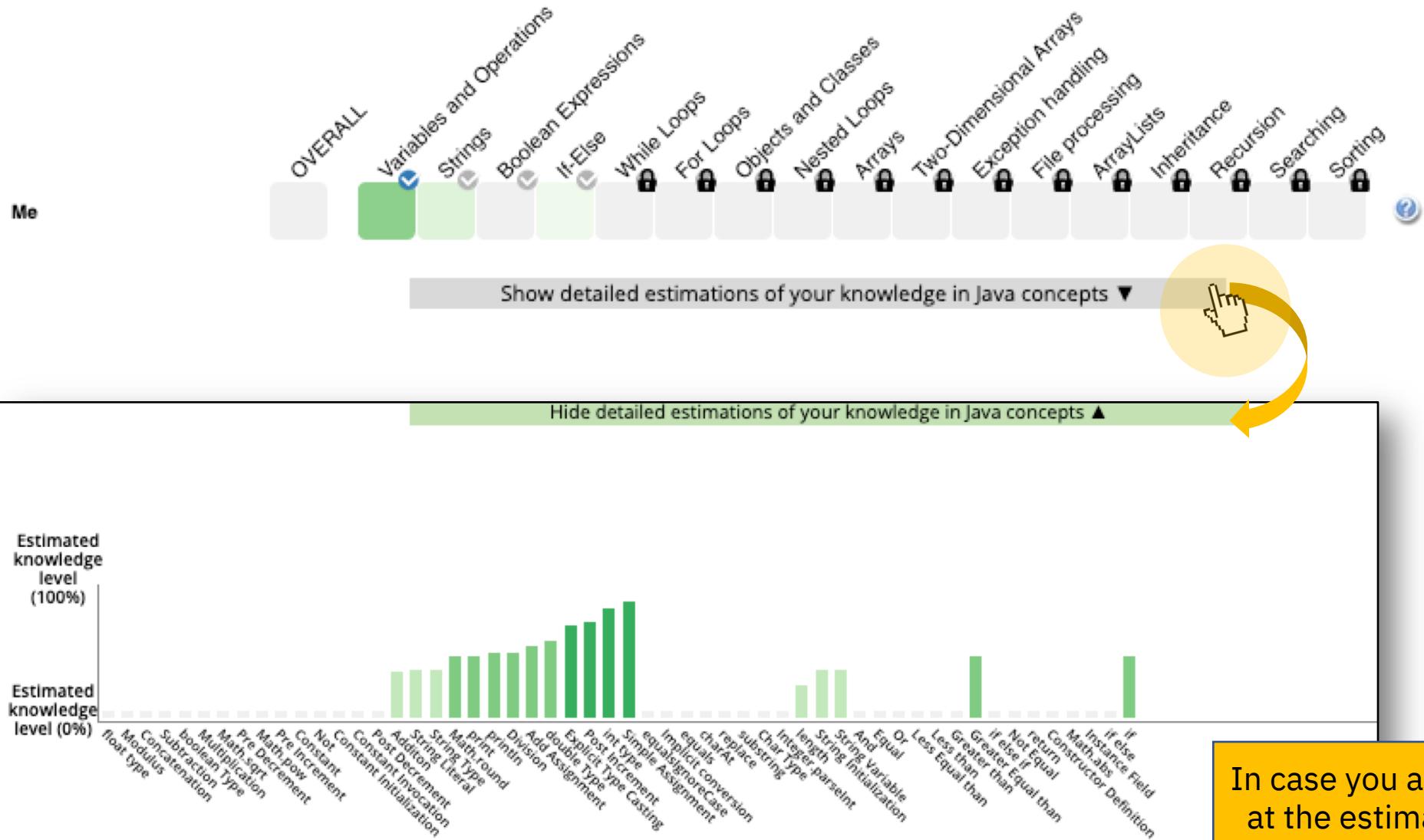
means that you got 1 point for completing at least 1 problem.

means that you have not completed any problem in this topic.

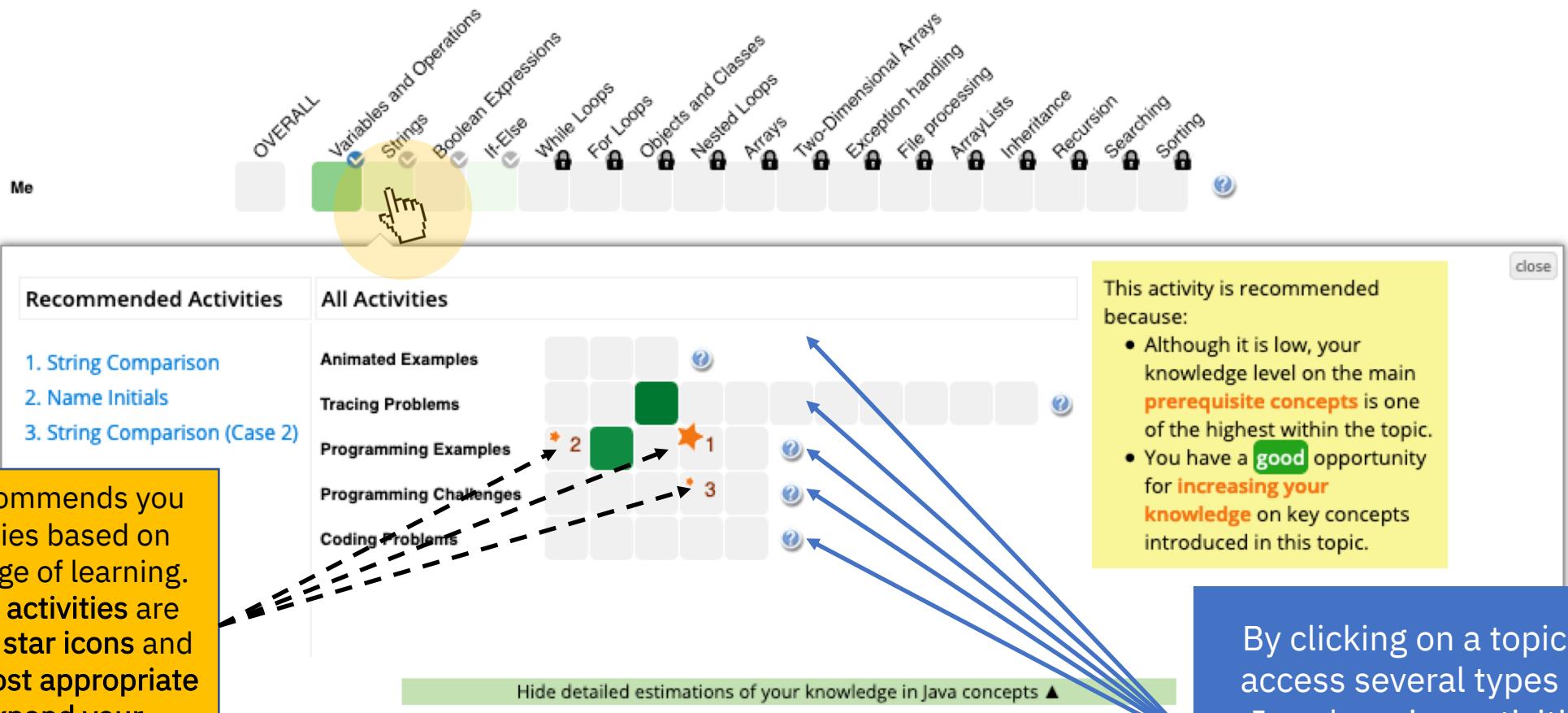
Topic Opening

means that the topic is not available for now but will be opened by your instructor at a later time.

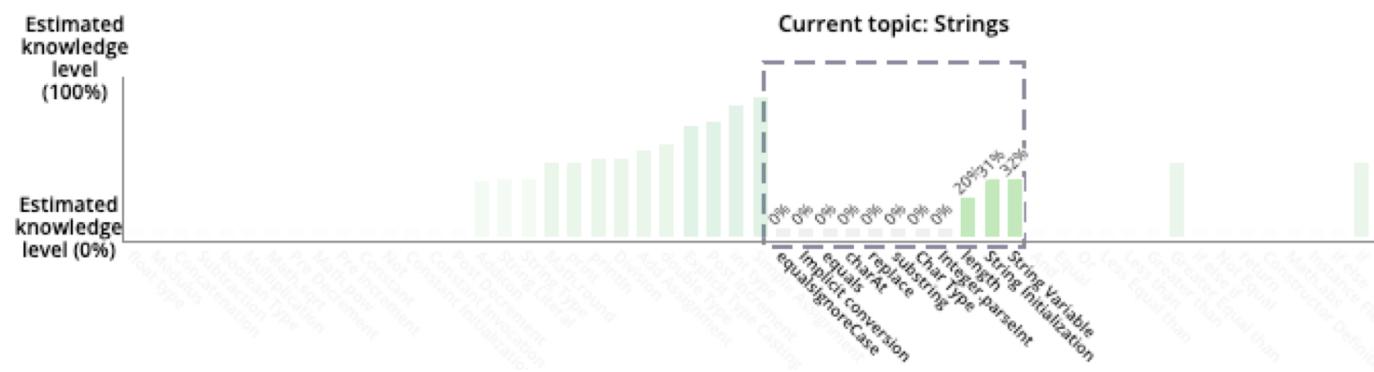
My Progress



In case you are curious, you can get a closer look at the estimation of proficiency the system has calculated for you in every fine-grained Java programming concept (click on "Show detailed ...")

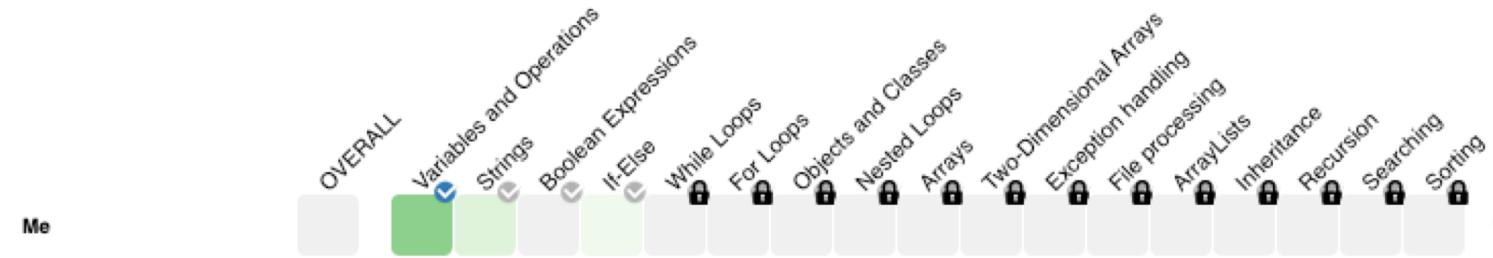


The system recommends you learning activities based on your current stage of learning. Recommended activities are highlighted with star icons and represent the most appropriate content to expand your knowledge.



By clicking on a topic cell, you can access several types of interactive Java learning activities which are represented by several grid cells.

My Progress



Recommended Activities

All Activities

1. String Comparison
2. Name Initials
3. String Comparison (Case 2)

Animated Examples

Tracing Problems

Programming Examples

Programming Challenges

Coding Problems

Hide detailed estimations of your knowledge in Java concepts ▲

A screenshot of the 'All Activities' section. It lists five activity types: Animated Examples, Tracing Problems, Programming Examples, Programming Challenges, and Coding Problems. The 'Programming Examples' row is highlighted with a yellow circle and a mouse cursor hovering over it. The 'Programming Challenges' row is also highlighted with a yellow circle. Each row contains three numbered items (1, 2, 3) and a question mark icon.

This activity is recommended because:

- Although it is low, your knowledge level on the main **prerequisite concepts** is one of the highest within the topic.
- You have a **good** opportunity for **increasing your knowledge** on key concepts introduced in this topic.

By mousing over a **recommended activity cell** you can have access to an **explanation on why** that specific content was recommended to you.

Also, in case you have activated the detailed view of concepts' proficiency estimations, you would be able to see the **concepts covered** by the moused over activity (this is available for all the activities – not only for recommended ones).

A detailed view of Java concepts' proficiency estimations. It shows a vertical bar chart with two horizontal axis labels: 'Estimated knowledge level (0%)' at the bottom and 'Estimated knowledge level (100%)' at the top. The chart displays bars for various concepts, with their estimated knowledge levels labeled above them. A dashed line indicates a trend or average. A callout box highlights the 'String' concept, which has an estimated knowledge level of 45%.

Animated examples

Topic: Strings • Activity: ToString Demo

```
1 public class ToStringDemo {  
2     public static void main(String[] args) {  
3         double d = 858.48;  
4         String s = Double.toString(d);  
5  
6         int dot = s.indexOf('.');  
7  
8         System.out.println(dot + " digits " + "before decimal point.");  
9         System.out.println((s.length() - dot - 1) + " digits after decimal point.");  
10    }  
11 }  
12 }  
13 }
```

Stack

Stack frame

Evaluation area

Text console

Literals

Double

Double.toString(d)

String

length() indexOf(c)

System.out.println(text)

+

-

Close window

Animated Examples (AE): AEs show visually how each step of an example problem is executed. These examples are useful to learn about the behavior of different programming constructs.

Tracing Problems

Topic: Strings • Activity: Strings 1

Tester

```
public class Tester {  
    public static void main(String[] args) {  
        String s = "what is the answer to this question?";  
        String sub = s.substring( 7, 11);  
        System.out.print(sub);  
    }  
}
```

What is the output?

Be careful of the space/newline in your answer.

Tracing Problems (TP): TPs assess your knowledge of how different programming contracts behave when being executed. If you feel that these problems are too hard, check [Animated Examples](#).

[Close window](#)

Tracing Problems (Table Trace)

Topic: Strings • Activity: Strings 1

Tester.java Trace Table

```
1+ public class Tester {  
2+   public static void main(String[] args) {  
3+     String s = "what is the answer to this question?";  
4+     String sub = s.substring( 7, 11);  
5+     System.out.print(sub);  
6+   }  
7+ }  
8  
9  
10
```

Tracing Problems (TP): TPs assess your knowledge of how different programming contracts behave when being executed. If you feel that these problems are too hard, check [Animated Examples](#).

In case you fail on getting the correct answer you will be able to trace the problem instance line by line to discover where you made the mistake and learn from it!

Trace Output

Line	Step	s	sub
3	2	what is the answer to this question?	
4	3		

Close window

Programming Examples

Topic: Strings • Activity: Name Initials

Example: Printing Name Initials

Construct a program that prints the initials of the name "John Smith".

```
1 public class Initials
2 {
3     public static void main(String[] args)
4     {
5         //Step 1: Define the name
6         String fullName = "John Smith"; ?
7         //Step 2: Extract the initial letter from the first name
8         String firstInitial = fullName.substring(0, 1); ?
9         //Step 3: Extract the initial letter from the last name
10        String lastInitial = fullName.substring(5, 6); ?
11        //Step 4: Create the initials from the initial letter in the
12        //first and the last name
13        String initials = firstInitial + lastInitial; ?
14        //Step 5: Print the initials
15        System.out.println(initials); ?
16    }
}
```

EXPLAIN THE PROGRAM 

Click on each line to get explanations

Programming Examples (PE): PEs walk you through complete solutions of meaningful programming problems. Use it to understand how programs should be constructed.

Close window

Programming Challenges

Topic: Strings • Activity: String With Escape Characters (Case 2)

Challenge: String With Escape Characters (Case 2)

Construct a program that prints the following output:

"Two Double Quotes"

Slashes \/

How "confounding" "\\" it is!

Drag a tile to each missing field to construct this program.

```
1 public class JEscapeChar2
2 {
3     public static void main (String[] args)
4     {
5         //Step 1: Print the first two lines
6
6
7         //Step 2: Print the last line
8         System.out.print("How '\"confounding' \"\\\" it is!");
9     }
10 }
```

Drag and drop the tiles to complete the right code

Drag a tile from here

CHECK

```
System.out.println("'''Two  
Double Quotes'''\tSlashes  
\\//");
```

```
System.out.println("\\\"\\\"Two  
Double Quotes\\\"\\n\\nSlashes  
\\\\\\\\");
```

```
System.out.println("\\"Two  
Double Quotes\"\\tSlashes  
\\\\\\\\");
```

[Close window](#)

Programming Challenges (PC): PCs assess your basic knowledge of problem construction. In these challenges, you are provided with a clear context and have to choose the correct programming construct to achieve the given goal.

Coding Problems

Topic: Strings • Activity: Concatenating Two Strings Without Including the First Character of Each of Them

Concatenating two strings without including the first character of each of them



Given two string variables `str1` and `str2`, complete the following code in order to concatenate the two strings without including the first character of each of them. Store the resulting string in a variable called `str3`. Assume that the initial value of variables `str1` and `str2` is already set to a string with at least one character.

E.g. 1: if the value of `str1` is "Hello" and the value of `str2` is "There", the code prints ellohere.

E.g. 2: if the value of `str1` is "java" and the value of `str2` is "code", the code prints avaode.

E.g. 3: if the value of `str1` is "shot!" and the value of `str2` is "java", the code prints hotlava.

```
1 | String str3;
2 | // TODO: add your code here
3 |
4 |
5 | System.out.println(str3);
```

Submit

Coding Problems (CP): CPs is the ultimate check of your program construction knowledge. Given the task, you need to write code to solve it. Your solution is checked using a set of tests.

Close window