CSE 121 – Lesson 8

Miya Natsuhara
Spring 2023

Music: 121 23sp Lecture Vibes 🌸

TAs:
Jasmine  Atharva  Mia  Justin
Shananda  Julia  Archit  Aishah
Vidhi  Anju  Grace  Claire
Larry  Lydia  Kailye  Lydia
Jacqueline  Jonus  Joshua  Kai
Afifah  Hugh  James

sli.do #cse121
Announcements, Reminders

• Programming Assignment 1 is due tonight, Wed April 26
• Creative Project 2 will be released later today
• Retakes
  • First round yesterday went smoothly!
  • Quiz 0 also eligible for retake on 5/2 and 5/9
• Quiz 1 scheduled for Thursday next week, May 2
• Wednesday May 3: Mid-term Formative Feedback
Common Problem-Solving Strategies

- **Analogy** – Is this similar to another problem you've seen?
- **Brainstorming** – Consider steps to solve problem before jumping into code
  - Try to do an example "by hand" → outline steps
- **Solve sub-problems** – Is there a smaller part of the problem to solve?
- **Debugging** – Does your solution behave correctly?
  - What is it doing?
  - What do you expect it to do?
  - What area of your code controls that part of the output?
- **Iterative Development** – Can we start by solving a different problem that is easier?
Metacognition

**Metacognition:** thinking about how you think
   Asking questions about your solution process

Examples

- **While debugging:** explain to yourself why you're making this change to your program
- **Before running your program:** make an explicit prediction of what you expect to see
- **When coding:** be aware of when you're not making progress, so you can take a break or try a different strategy
- **When studying:** What is the relationship of this topic to other ideas in the course?
Returns allow us to send values out of a method

```java
public static <type> myMethod(int num) {
    System.out.print(num + " is the best!");
    ...
    return <value of correct type>
}
```

Calling a method that returns a value...

```java
<type> result = myMethod(42);
```
### (Recall) String Methods

Usage: `<string variable> . <method>(...)`

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>length()</code></td>
<td>Returns the length of the string.</td>
</tr>
<tr>
<td><code>charAt(i)</code></td>
<td>Returns the character at index <code>i</code> of the string</td>
</tr>
<tr>
<td><code>indexOf(s)</code></td>
<td>Returns the index of the first occurrence of <code>s</code> in the string; returns -1 if <code>s</code> doesn't appear in the string</td>
</tr>
<tr>
<td><code>substring(i, j)</code> or <code>substring(i)</code></td>
<td>Returns the characters in this string from <code>i</code> (inclusive) to <code>j</code> (exclusive); if <code>j</code> is omitted, goes until the end of the string</td>
</tr>
<tr>
<td><code>contains(s)</code></td>
<td>Returns whether or not the string contains <code>s</code></td>
</tr>
<tr>
<td><code>equals(s)</code></td>
<td>Returns whether or not the string is equal to <code>s</code> (case-sensitive)</td>
</tr>
<tr>
<td><code>equalsIgnoreCase(s)</code></td>
<td>Returns whether or not the string is equal to <code>s</code> ignoring case</td>
</tr>
<tr>
<td><code>toUpperCase()</code></td>
<td>Returns an uppercase version of the string</td>
</tr>
<tr>
<td><code>toLowerCase()</code></td>
<td>Returns a lowercase version of the string</td>
</tr>
</tbody>
</table>
String example

String s = "gumball";

s = s.substring(7, 8).toUpperCase() + s.substring(8) + "ball";
## Example of returns: Math class

<table>
<thead>
<tr>
<th>Methods</th>
<th>Returns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Math.abs(value)</td>
<td>Absolute value of value</td>
</tr>
<tr>
<td>Math.ceil(value)</td>
<td>value rounded up</td>
</tr>
<tr>
<td>Math.floor(value)</td>
<td>value rounded down</td>
</tr>
<tr>
<td>Math.max(value1, value2)</td>
<td>Larger of the two given values</td>
</tr>
<tr>
<td>Math.min(value1, value2)</td>
<td>Smaller of the two given values</td>
</tr>
<tr>
<td>Math.round(value)</td>
<td>value rounded to the nearest whole number</td>
</tr>
<tr>
<td>Math.sqrt(value)</td>
<td>Square root of value</td>
</tr>
<tr>
<td>Math.pow(base, exp)</td>
<td>base to the exp power</td>
</tr>
</tbody>
</table>
Math example

double value = 823.577564893;
double roundedValue = (double) Math.round(value * 100) / 100;
What is the correct implementation of a `maxDatingAge` method?

A. ```java
public static void maxDatingAge(int age) {
    int maxDatingAge = age - 7 * 2;
    return maxDatingAge;
}
```  

B. ```java
public static void maxDatingAge(int age) {
    int maxDatingAge = age - 7 * 2;
}
```  

C. ```java
public static int maxDatingAge(int age) {
    int maxDatingAge = (age - 7) * 2;
    return maxDatingAge;
}
```  

D. ```java
public static int maxDatingAge(int age) {
    return (age - 7) * 2;
}
```
Poll in with your answer!

What is the output of this program?

```java
public static int returnExample() {
    for (int i = 0; i < 5; i++) {
        return i;
    }

    return -1;
}
```

A. 0
B. 4
C. 5
D. -1