

## **Debugging Quiz**

### **Description**

During Week 6 (starting Monday 3/3), we will have quiz 2 (the debugging quiz). **You will not have in-person lab sessions on Mon. (3/3) and Wed. (3/5) of Week 6.** Instead, you will show up at a predetermined time to meet with a TA one on one to take the quiz. You must bring your laptop.

A TA (not necessarily your teaching TA) will contact you this week with the time of your quiz. Please make sure you check your terpmail University email account for the email from the TA with your quiz time. The TAs will schedule your quiz during your Mon. or Wed. lab time, which are times that you should have reserved for CMSC 132. If you have a valid reason for why you cannot be available at your assigned time, you can discuss that with the TA **AFTER** you receive the email with your time.

**BE ON TIME. TAs will not wait around for you and need to go to the next person. If you don't show up on time, the TA has no obligation to give you full time or the quiz at all. Wait outside your discussion classroom until the TA calls you in.**

At the end of Week 5, I will post the code for the debug quiz and some pre-quiz time instructions. Follow the instructions so that you can be ready for the quiz at the start of Week 6. Based on the time you get; your quiz may be as early as first time slot on Monday 3/3 (i.e. first discussion time) or as late as Wed. 3/5 at end of your lab time. **If you have any questions about debugging, please ask/post on Piazza this week. Once the code is distributed at the end of Week 5, we will not answer any questions about using the debugger until the quiz is over.** As students are taking the quiz at different times, I ask that you please do not discuss or post about the contents of the quiz until all students have taken it.

To prepare for the quiz, you should be able to do the following things:

1. Understand and be able to switch from the Java and Debug Perspectives and run the code in each
2. Set line numbers by right-clicking on the left column of the code page.
3. Set and remove breakpoints.
4. Step into, step over, and step return from a method.
5. Display variables, parameters, and member field values.
6. Continue program execution after a breakpoint and terminate a debug session.
7. Switch between the stack frame of one method to another.

### **Resources**

#### **1. Debugger Lecture Videos**

##### **Fall 2020**

<https://umd.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=7583722e-4d20-4502-a333-abd400960e32> – (starting at 34:00)

<https://umd.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=b3847176-0bea-4295-99ac-abd50015351e> – (start to 6:45)

##### **Fall 2021**

<https://umd.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=aaa725e0-e076-44f7-8466-ada901081991> – (36:15 to end)

#### **2. Code used in Debugger Lecture Video**

<https://www.cs.umd.edu/class/spring2025/cmsc132-010X-020X/lectures/Week5/DebuggerExampleCode.zip>

#### **3. Additional outside resource**

<http://www.vogella.com/tutorials/EclipseDebugging/article.html>