|  |  |  |
| --- | --- | --- |
| **Date: 11/30** | **SI Facilitator: Christopher Simon** | **SI Course: CSC 15** |

**Objective(s):**What are the most important concepts that the students need to work on today?

* Going over practice problems for the second midterm.
* Understanding how to set up problems and breaking them down by solvable pieces.
* Learning how to translate the requirements from English as sudo code and then into actual code.

**Opening technique(s):**

|  |  |  |
| --- | --- | --- |
| **Time** | **Content to be Covered** | **Description of Activity/Process** |
| 3:00 | Getting caught up with the class on both chapters 6 and 8 | In class, we spent the first few minutes making sure that everyone was aware that both chapters 6 and 8 were going to be due by the end of the week, even though chapter 6 would not be on the final. After asking the class, we decided to go over some of the problems from chapter 6, as most students were unfamiliar with the content covered in class for it. |
| 3:05 | Reviewing chapter 6 | Before going over some of the problems, as per request, I went over the basics of file reading and saving. Since we didn’t have time to cover this back when I took CSc 15 and because there are quite a few different ways of going about this, I spent a few minutes orienting myself with the specific method taught in the course. From there I was able to successfully get the basics across. |
| 3:25 | Going over one of the practice it problems | We ended up going over one of the practice problems from the practice it. I tried to get the students involved in helping write the code, though this didn’t seem to work out as well, mainly because the material was too new to the students. I emailed them the code in the end as usual. |

**Why did you implement these activities and process?**

* Going over new material is important. I wouldn’t have been able to go over some of the problems if the students didn’t understand anything about the chapter.
* Involving the class in helping me write the solution usually works, hence why I tried it again this time.

**Reflections on effectiveness of activity/process, challenges, insights to improve for the future and assess how you are developing as a SI leader:**

* I really would have like to have gotten the class even more involved than they were, however I’m not too sure what else I could have done in order to make the material more understandable and engaging.

1 // Import these whenever we are working with Files  
 2 import java.io.\*;  
 3 import java.util.Scanner;  
 4   
 5 public class Problem613 {  
 6 public static void main(String[] args) {  
 7 // Whever we are dealing with files, we need to use this try catch block.  
 8 // Just memorize it for now -- You'll learn more about it in CSc 20.  
 9 try {  
10   
11 // Load the file  
12 File f = new File("code.java");  
13   
14 // Create a scanner for the loaded file.  
15 // This scanner will help us to traverse each line of the file  
16 Scanner file = new Scanner(f);  
17   
18 // Strip the comments from the file  
19 stripComments(file);  
20   
21 } catch(FileNotFoundException e) {  
22 // We can't find the file. Print out an error.  
23 System.err.println(e);  
24 }  
25 }  
26   
27 public static void stripComments(Scanner file) {  
28 // Keep running until there are no more lines in the file  
29 while(file.hasNext()) {  
30 // Get the next line from the file.  
31 // This is something that we've used in the past.  
32 // file is a scanner, so we can use methods like  
33 // nextLine(), nextInt() and next() to read the data from it,  
34 // just like we do for the keyboard.  
35   
36 // Hint: Use substring to check for "/\*" and "\*/" remove everything in between  
37 // Do this for "//" as well   
38   
39 // Print out the next line in the file  
40 System.out.println( file.nextLine() );  
41 }  
42 }  
43 }