-DebuggingExercise.java:

The code was creating an array with 4 positions and the for loops where going out of bounds because the starting value of I was “1” and the limit was “<5”, this error is easy to detect once we locate breakpoints on lines “9” and “16” to trace the changes on the index of the array. Once the error is detected, we just change the initial value as “0” and the limit as “4”, keeping the looping process inside the array [0,1,2,3].

-DebugHash.java:

To get the 49,791st hash value, we introduce a breakpoint and we introduce a hit count, for this case we need to stop one value before the 49,791(49,790) because the 49,790th position in the array saves the 49,791st hash value. During the debugging process we should put just one breakpoint, this break point will be located at line “40” where the “I” variable (the index variable or counter) in order to keep track of the index position in the array. In order to stop in the 49,790 position we will select the breakpoint properties to put the desired position in “Hit count”, after debugging the infinite loop will stop in the desired position, in this case the 49,791st hash value is “ffcf2d0a8af9e938de4e1b7528a1dd03”.

-FibDebug.java:

In this code there are no syntax errors, so we can run perfectly the code. The problem is inside the while condition, the “n” value is limited to be bigger than 1 skipping the las step in the Fibonacci series. After we change that condition our code is free of errors.

-Marker.java:

After putting a series of break points over the if conditions and proceeding step by step, we can tell that the if conditions are bad stated, in this case 90 satisfies all the conditions. After a few changes in the condition we can run the code and see that the range of the grades are linked just with one of the print messages

-Account/AccountDebug.java:

After running and debugging the code with a point exception in the constructor and the call of the constructor we can see that the constructor is called incorrectly, after the call of the constructor is fixed with the appropriate structure and following the constructor requirements “Account a = **new** Account(**null**);” , we will be able to run the code properly.

-Person/PersonDebug.java:

In this case the code is full of basic syntax errors which need to be fixed. After that, we can see that the constructor does not follow the proper structure so we fix it changing the name (just the lower case letter) and the initializations using “this.”. After those changes we try the code and we can see that the print message calls twice the “getName” method, after switching the second one for a “getAge” we have a good java file.