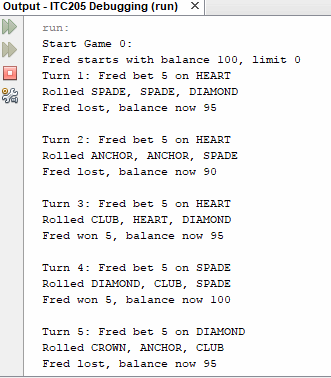
# Bug 4 – Debugging Log

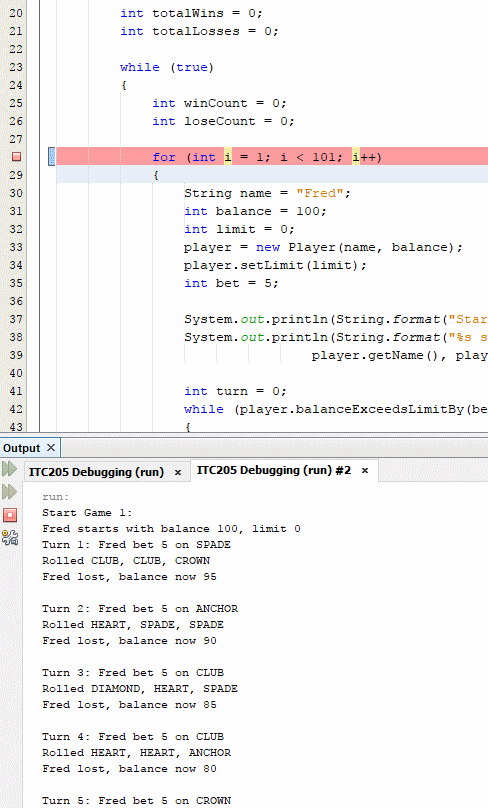
This bug consists of 2 unreported lesser bugs that affect the text output of the program. These are not programmatic errors and do not affect the functionality of the program, but they do detract from the user experience.

These bugs are already in their simplest form when running the program via the Main class, as the bugs are present in that class. For this reason I did not create an automated test, as the test would run the main class and then be unable to assert anything. Before the fixes, the main output looked like this:



The first of these bugs is that the first game played by the program is listed as Game 0. This in turn has a carry-on effect as all games are listed to be one lower than the actual game they should be numbered as. This bug is labelled as Bug 4a.

To fix this, I changed the loop running the games to initialise its iterator value at 1 instead of 0, and increased the maximum number for it by 1 to compensate. After screenshot is below.



The second bug is that the DiceValues are output in all capital letters. While fixing other bugs, I noticed that there was code in place to convert a DiceValue to a String that had only the first letter capitalised, but this code was never used.

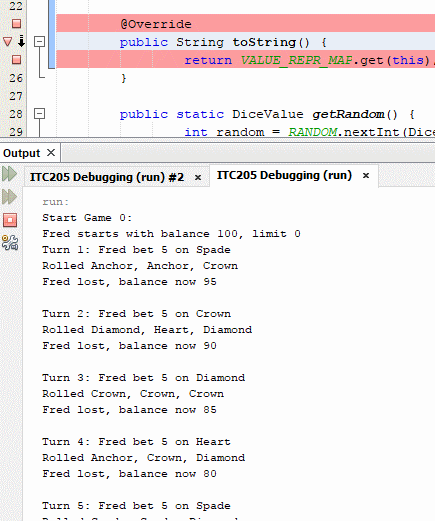
The easy fix to this would be to call this toString(DiceValue value) method in the main class when a line is printed. However, since toString is the only non-static method in the DiceValue class, each call to that method would look like: value.toString(value) . This doesn’t look quite right, so I decided to start looking into other solutions.

These DiceValues are displayed as arguments to a System.out.Printf() call, using the format specifier “%s”. Looking at the Javadoc for this method, I found that the ‘s’ in ‘%s’ indicated a general conversion that, if the argument is not null or does not implement Formattable, it obtains a result by calling argument.toString(). This is the Java.Lang.Object.toString() method, which is overridden by the Java.Lang.Enum to return the name of the enum value.

So by overriding this toString() method with the one in the DiceValue class, the correct, uncapitalized string will be returned when a DiceValue is used as an argument in System.out.Printf() call.

The exact changes to make to this method to accomplish this are: to remove the value passed to the toString method, to use ‘this’ (the object on which toString() is being called) to get the string from the VALUE\_REPR\_MAP, and to add the “@Override” annotation to the method.

The after screenshot is below.



After fixing both bugs, the output looks like this:

