# Bug 1 – Debugging Log

The bug essentially boils down to the player’s “Balance” variable being set to an incorrect value over the course of a round. So we look at the Player class to see which methods set or change that variable.

The three methods that set the Balance are the constructor, the takeBet() method, and the recieveWinnings method. I suspect the problem may lie in one of the two latter methods, so I perform usage searches on them both.

These two methods are used in code exactly once each: in the Game.playRound() method. The takeBet method is called at the start of this method, and the recieveWinnings method is called at the end if there is more than 0 matches.  
  
At this point I added a few lines of code to the playRound method that simply printed the player’s balance to the system output at specific points in the code. This is the output of my test after adding these checks:

------------- Standard Output ---------------

playRound

John's balance is 100

John bet 5 on HEART

Rolled HEART, CROWN, ANCHOR

Player balance before taking bet: 100

Player balance after taking bet, before recieving winnings: 95

Player balance after recieving winnings: 100

John won 5, balance now 100

------------- ---------------- ---------------

Testcase: testPlayRound(unit\_test.TestBug1): FAILED

expected:<105> but was:<100>

These checks demonstrated that both the takeBet() and recieveWinnings() methods were working as intended, but that calling them both resulted in the bugged values.

I quickly realised that the bug was due to the player’s bet being taken even when they won. The intended behaviour is that the player will keep their bet if they win.

To fix the bug, the simplest method would be to add the value of the bet to the winnings in line 49 of the game class. And the simplest method is often the best. Here are screenshots of the code, before and after:



