# Hypotheses and Tests

## Bug 1 Hypothesis 1:

The player, after a win, is not receiving either their win or their original bet back. E.g. if the bet 5 and win 5 they should get 10 back not 5 as the code is doing. So either the bet or winnings is not being added to the balance upon a win.

## Bug 1 Hypothesis 1 Test (RESOLUTION):

Inspected a method in the Player.java file that increases the player balance for each win. “receiveWinnings” was not accepting the initial bet. So adding “**int** bet” to “**public** **void** receiveWinnings(**int** winnings)” and “+ bet” to the equation “balance = balance + winnings;” resulted in an error occurring in Game.java.

Due to this new error, I then added “bet” to “player.receiveWinnings(winnings);”. After running these changes, it appeared that the bug was resolved.

## Bug 2 Hypothesis 1:

I think that the program is only accepting values over the limit and not the limit due to a syntactical error. The limit should be accepting the balance on and over the limit amount.

## Bug 2 Hypothesis 1 Test (RESOLUTION):

By searching for where the player’s balance exceeds the limit in the Player.java file. The method “balanceExceedsLimitBy” contains the error. Changing “**return** (balance - amount > limit);” to “**return** (balance - amount >= limit);” and re-running the program has resolved the bug.

## Bug 3 Hypothesis 1:

The ratio of win and loses are not being calculated correctly.

## Bug 3 Hypothesis 1 Test (FAILED):

By changing “System.***out***.println(String.*format*("Overall win rate = %.1f%%", (**float**)(totalWins \* 100) / (totalWins + totalLosses)));” in Main.java to “(**float**)((totalWins) / (totalWins + totalLosses) \* 100)));”

After running the program with this change, it did not appear to resolve the win/lose ratio to 0.42.

## Bug 3 Hypothesis 2:

A value in Dice.java is incorrect.

## Bug 3 Hypothesis 2 Test (FAILED):

Changed “roll()” in Dice.java to return “value”. i.e. “**return** DiceValue.*getRandom*();” to “**return** value = DiceValue.*getRandom*();”. After running this change, it did not seem fix this bug.

## Bug 3 Hypothesis 3:

A value in DiceValue.java is incorrect.

## Bug 3 Hypothesis 3 Test (RESOLVED):

In combination with Hypothesis 2, I changed equation (in the method “getRandom()” of DiceValue.java) “**int** random = *RANDOM*.nextInt(DiceValue.***SPADE***.ordinal());” to add 1, so it then computes “**int** random = *RANDOM*.nextInt(DiceValue.***SPADE***.ordinal() + 1);”.After running this change, the win/lose ratio seems to appear a constant 0.42 thus resolving this bug.