**IT6727- Summary Report**

**Arshdeep Singh (20231463)**

**Git Hub Repository Link -**<https://github.com/arshdeep26109/SoftwareTesting-Assignment2>

**Report Contents**

[Part A - Report](#_fjmded3s4dch)

[Part B: Test Log](#_skb7ssmewtjr)

[Part C - Testing Screenshots](#_tzrifhgb3965)

# Part A - Report

The assignment required to test and maintain the bowling game code. The provided file was not scoring correct in all cases. For this solution I created six test cases. I named them tc01.py up to tc06.py. The test cases were standard, covering gutter, perfect, all spares, normal open, spare in the last frame and strike in the last frame. Each test was a small script that made rolls and printed the expected score and the actual one.

The first test tc01 for gutter game was fine, it showed 0 both times. The second test for perfect game showed a fail. The expected score was 300 but the actual output was 270. This was a clear bug. I checked the scoring logic and found that the strike bonus did not add the next two balls correctly in sequence. I changed the code so the bonus calculation includes the right indexes. After this the test passed with 300. This was committed to git.

Test tc03 was all spares, expected 150, actual was also 150 but tc04 was regular open game, this gave wrong number 38 instead of 72. I saw that the code only counted one roll in the open frame. It missed the second roll. I refactored by adding both rolls together. This gave the right output. A commit was made for this.

The last two cases tc05 and tc06 were for final frame spare and strike. Both worked as expected, so no change was required. That confirmed the fixes were enough and no more major defects exist.

Hence, the refactoring was two main things. Fix strike bonus indexing. Fix open frame total calculation. These two bugs made the score wrong in some games. After fixing, all cases pass. The program is now much more reliable. The improvement is that now the implementation follows bowling business rules exactly, which is critical since this is an educational software.

I also generated PythonDoc. The HTML files explain each function with docstrings. This will help in future work, like when GUI is added. Git commits were done. The main learning is that testing small parts with unit tests is a good way to avoid expensive mistakes in later development stages.

The bowling backend is correct and can be used in the next development stage.

# Part B: Test Log

| **Test ID** | **Test Case** | **Expected** | **Actual** | **1st Result** | **Refactoring Done** | **Final Result** |
| --- | --- | --- | --- | --- | --- | --- |
| TC01 | Gutter Game | 0 | 0 | Pass | No | Pass |
| TC02 | Perfect Game | 300 | 270 | Fail | Fixed | Pass |
| TC03 | All Spares | 150 | 150 | Pass | No | Pass |
| TC04 | Regular Game | 72 | 38 | Fail | Fixed | Pass |
| TC05 | Spare Last Frame | 15 | 15 | Pass | No | Pass |
| TC06 | Strike Last Frame | 30 | 30 | Pass | No | Pass |

# Part C - Testing Screenshots