# Withdrawal Test Case Generation

The Slice for FinalUse(fee) is the entire function calculateWithdrawalFee. This is because any variable declarations or assignments and conditionals that directly affect fee must be included.

For statement coverage we need test cases that cover each path, since there is only 5 paths 5 test cases should suffice.

The test cases will cover:

Student status: true or false

Day of the week: weekend or weekday

Account balance: less than $1000, between $1000 and $10,000, and $10,000 and more

Tests:

Case 1: {student, weekend, 999}

Case 2: {student, week day, 999}

Case 3: {non student, weekend, 999}

Case 4: {non student, weekend, 1000}

Case 5: {non student, weekend, 10001}

Expected Results:

Case 1: 0.0

Case 2: 0.001

Case 3: 0.002

Case 4: 0.001

Case 5: 0.0

# Transfer Test Case Generation

Process and Assumptions to generate test cases.

We used the McCabe algorithm to determine our basis paths. We looked at the code and chose a path through the logic all the way to the end, that was our first path. Then we proceeded to move backward from the end, flipping conditions as we went to generate our basis paths.

We assumed that the numbers for the Transfer fee percentage were to be divided by a hundred before being used to calculate the fee. So if the fee was 0.1% in the pdf document, then our calculation would be amount \* (0.1 / 100).

Tables with Test Cases

A screenshot of a table

Description automatically generated

Test Results

A screenshot of a table

Description automatically generated