# Test Cases for Objective-C

## Test Case 1

### Test Objectives

To check how the system respond if no argument is provided for execution.

### Test Description

Just Insert UCC (without any argument).

### Pre-conditions

Test environment is ready to use. (i.e.no compilation error).

### Post-conditions

System should not crash.

### Dependencies

No dependencies.

### Assumptions and Constraints

The code is well written to take care of any side effects caused by the test.

### Input Specifications

Insert UCC from command prompt for windows environment

### Expected Output Specifications

An error message stating that “No arguments found” and if possible system should suggest how to provide arguments.

### Pass/Fail Criteria

Expected output should be shown.

### Test Results

PASS

## Test Case 2

### Test Objectives

To verify the system reads multiple files for generating code count output.

### Test Description

Insert UCC –dir src \*.m command.

### Pre-conditions

Test environment is ready to use. (i.e.no compilation error).

### Post-conditions

System should not crash

### Dependencies

No dependencies.

### Assumptions and Constraints

The code is well written to take care of any side effects caused by the test.

### Input Specifications

Insert “UCC –dir src \*.m” in command prompt for windows environment

### Expected Output Specifications

Output files “outfile\_cplx.csv” & “M\_outfile.csv” should be generated with correct line counts for multiple files.

### Pass/Fail Criteria

Expected output should be shown.

### Test Results

PASS

## Test Case 3

### Test Objectives

To find out which lines are counted as compiler directive in Objective C.

### Test Description

Insert UCC –dir src \*.m command

### Pre-conditions

Test environment is ready to use. (i.e.no compilation error).

### Post-conditions

System should not crash

### Dependencies

No dependencies

### Assumptions and Constraints

The code is well written to take care of any side effects caused by the test.

### Input Specifications

Insert “UCC –dir src \*.m” in command prompt for windows environment

### Expected Output Specifications

The output fine should contain the number of the compiler directives that the source contains.

**3.1.9 Pass/Fail Criteria**

Expected output should be shown.

### Test Results

PASS

## Test Case 4

### Test Objectives

To check how many lines of logical SLOC the code counter counts for a ‘for’ statement.

### Test Description

A piece of code containing two for loops is given

### Pre-conditions

Test environment is ready to use(i.e. no compilation error).

### Post-conditions

System should not crash.

### Dependencies

No dependencies.

### Assumptions and Constraints

The code is well written to take care of any side effects caused by the test.

### Input Specifications

Insert UCC from command prompt for windows environment

### Expected Output Specifications

The lines of code should be two for the two ‘for’ loops and the remaining lines of code for the remaining part of the code.

### Pass/Fail Criteria

Expected output should be shown.

### Test Results

PASS

## Test Case 5

### Test Objectives

When the code counter encounters a while block, it is accounted for one logical SLOC.

### Test Description

Code count a code with a while loop

### Pre-conditions

Test environment is ready to use. (i.e. no compilation error).

### Post-conditions

System should not crash.

### Dependencies

No dependencies.

### Assumptions and Constraints

The code is well written to take care of any side effects caused by the test.

### Input Specifications

Run the input test code for the test case.

### Expected Output Specifications

One count for the single while loop is added to the total count.

### Pass/Fail Criteria

Expected output should be shown.

### Test Results

PASS

## Test Case 6

### Test Objectives

A do-while loop counts for a single LOC.

### Test Description

When the code count encounters one do-while loop, the SLOC increases by 1.

### Pre-conditions

Test environment is ready to use. (i.e.no compilation error).

### Post-conditions

System should not crash.

### Dependencies

No dependencies.

### Assumptions and Constraints

The code is well written to take care of any side effects caused by the test.

### Input Specifications

Run the input test code for this test case.

### Expected Output Specifications

The number of SLOC in the code should be outputted.

### Pass/Fail Criteria

Expected output should be shown.

### Test Results

PASS

## Test Case 7

### Test Objectives

An if statement counts as one logical SLOC.

### Test Description

A test case consisting of an if statement.

### Pre-conditions

Test environment is ready to use. (i.e.no compilation error).

### Post-conditions

System should not crash.

### Dependencies

No dependencies.

### Assumptions and Constraints

The code is well written to take care of any side effects caused by the test.

### Input Specifications

Run the input test code for this case

### Expected Output Specifications

The expected number of SLOC should be outputted.

### Pass/Fail Criteria

Expected output should be shown.

### Test Results

PASS

## Test Case 9

### Test Objectives

To count the number of blank lines in the code.

### Test Description

To insert code that consists of some blank lines and some code.

### Pre-conditions

Test environment is ready to use. (i.e.no compilation error).

### Post-conditions

The blank lines should not be counted by the code counter.

### Dependencies

No dependencies.

### Assumptions and Constraints

The code is well written to not count the blank lines in the logical lines of code and also count the number of blank lines.

### Input Specifications

Insert UCC from command prompt for windows environment

### Expected Output Specifications

The number of blank lines are displayed.

### Pass/Fail Criteria

Expected output should be shown.

### Test Results

PASS

## Test Case 9

### Test Objectives

To count the number of commented lines in the entire code.

### Test Description

Insert the code to be tested in the code counter.

### Pre-conditions

Test environment is ready to use. (i.e.no compilation error).

### Post-conditions

The commented lines should not be counted in the logical lines of code and we should have a separate count of the commented lines.

### Dependencies

No dependencies.

### Assumptions and Constraints

The code gives a separate count of the commented lines of code.

### Input Specifications

Insert UCC from command prompt for windows environment.

### Expected Output Specifications

The number of commented lines are outputted.

### Pass/Fail Criteria

The correct number of commented lines should be shown in the output.

### Test Results

PASS

## Test Case 10

### Test Objectives

A Try Catch statement counts as one logical SLOC.

### Test Description

A test case consisting of a Try Catch statement.

### Pre-conditions

Test environment is ready to use. (i.e.no compilation error).

### Post-conditions

System should not crash.

### Dependencies

No dependencies.

### Assumptions and Constraints

The code is well written to take care of any side effects caused by the test.

### Input Specifications

Run the input test code for this case

### Expected Output Specifications

The expected number of SLOC should be outputted.

### Pass/Fail Criteria

Expected output should be shown.

### Test Results

PASS

## Test Case 11

### Test Objectives

To check if the system takes empty statement as one logical SLOC.

### Test Description

A test case consisting of an empty statement.

### Pre-conditions

Test environment is ready to use. (i.e.no compilation error).

### Post-conditions

System should not crash.

### Dependencies

No dependencies.

### Assumptions and Constraints

The code is well written to take care of any side effects caused by the test.

### Input Specifications

Run the input test code for this case

### Expected Output Specifications

Correct number of SLOC should be counted for an empty statement.

### Pass/Fail Criteria

Expected output should be shown.

### Test Results

PASS

## Test Case 12

### Test Objectives

### An return statement counts as one logical SLOC.

### Test Description

A test case consisting of a return statement.

### Pre-conditions

Test environment is ready to use. (i.e.no compilation error).

### Post-conditions

System should not crash.

### Dependencies

No dependencies.

### Assumptions and Constraints

The code is well written to take care of any side effects caused by the test.

### Input Specifications

Run the input test code for this case

### Expected Output Specifications

The expected number of SLOC should be outputted.

### Pass/Fail Criteria

Expected output should be shown.

### Test Results

PASS

## Test Case 13

### Test Objectives

An exit statement counts as one logical SLOC.

### Test Description

A test case consisting of an exit statement.

### Pre-conditions

Test environment is ready to use. (i.e.no compilation error).

### Post-conditions

System should not crash.

### Dependencies

No dependencies.

### Assumptions and Constraints

The code is well written to take care of any side effects caused by the test.

### Input Specifications

Run the input test code for this case

### Expected Output Specifications

The expected number of SLOC should be outputted.

### Pass/Fail Criteria

Expected output should be shown.

### Test Results

PASS

## Test Case 14

### Test Objectives

When GOTO statements are encountered, the lines skipped are counted as physical lines of code

### Test Description

A test case consisting of a goto statement.

### Pre-conditions

Test environment is ready to use. (i.e.no compilation error).

### Post-conditions

System should not crash.

### Dependencies

No dependencies.

### Assumptions and Constraints

The code is well written to take care of any side effects caused by the test.

### Input Specifications

Run the input test code for this case

### Expected Output Specifications

The expected number of SLOC should be outputted. The physical SLOC count is incremented by even for the skipped lines

### Pass/Fail Criteria

Expected output should be shown.

### Test Results

PASS

## Test Case 15

### Test Objectives

When the code counter encounters a semicolon directly following a for loop,it is counted as 2 logical SLOC

### Test Description

A test case consisting of a semicolon statement.

### Pre-conditions

Test environment is ready to use. (i.e.no compilation error).

### Post-conditions

System should not crash.

### Dependencies

No dependencies.

### Assumptions and Constraints

The code is well written to take care of any side effects caused by the test.

### Input Specifications

Run the input test code for this case

### Expected Output Specifications

The logical SLOC count is incremented by 2 when the for (); statements is encountered

### Pass/Fail Criteria

Expected output should be shown.

### Test Results

PASS

## Test Case 16

### Test Objectives

When the code count tool encounters an IF..ELSE block .The logical SLOC is 1 for the if..else statements, if and else are not counted seperately

### Test Description

A test case consisting of a if else statement.

### Pre-conditions

Test environment is ready to use. (i.e.no compilation error).

### Post-conditions

System should not crash.

### Dependencies

No dependencies.

### Assumptions and Constraints

The code is well written to take care of any side effects caused by the test.

### Input Specifications

Run the input test code for this case

### Expected Output Specifications

The expected number of SLOC should be outputted.

### Pass/Fail Criteria

Expected output should be shown.

### Test Results

PASS

## Test Case 17

### Test Objectives

Files with .m extension are counted using objective-C counting standards

### Test Description

A file with .m extension.

### Pre-conditions

Test environment is ready to use. (i.e.no compilation error).

### Post-conditions

System should not crash.

### Dependencies

No dependencies.

### Assumptions and Constraints

The code is well written to take care of any side effects caused by the test.

### Input Specifications

Run the input test code for this case

### Expected Output Specifications

Count of SLOC for objective-C files is returned

### Pass/Fail Criteria

Expected output should be shown.

### Test Results

PASS

## Test Case 18

### Test Objectives

An switch statement counts as one logical SLOC.

### Test Description

A test case consisting of a switch statement.

### Pre-conditions

Test environment is ready to use. (i.e.no compilation error).

### Post-conditions

System should not crash.

### Dependencies

No dependencies.

### Assumptions and Constraints

The code is well written to take care of any side effects caused by the test.

### Input Specifications

Run the input test code for this case

### Expected Output Specifications

The expected number of SLOC should be outputted.

### Pass/Fail Criteria

Expected output should be shown.

### Test Results

PASS

## Test Case 19

### Test Objectives

An break statement counts as one logical SLOC.

### Test Description

A test case consisting of an break statement.

### Pre-conditions

Test environment is ready to use. (i.e.no compilation error).

### Post-conditions

System should not crash.

### Dependencies

No dependencies.

### Assumptions and Constraints

The code is well written to take care of any side effects caused by the test.

### Input Specifications

Run the input test code for this case

### Expected Output Specifications

The expected number of SLOC should be outputted.

### Pass/Fail Criteria

Expected output should be shown.

### Test Results

PASS

## Test Case 20

### Test Objectives

An continue statement counts as one logical SLOC.

### Test Description

A test case consisting of an continue statement.

### Pre-conditions

Test environment is ready to use. (i.e.no compilation error).

### Post-conditions

System should not crash.

### Dependencies

No dependencies.

### Assumptions and Constraints

The code is well written to take care of any side effects caused by the test.

### Input Specifications

Run the input test code for this case

### Expected Output Specifications

The expected number of SLOC should be outputted.

### Pass/Fail Criteria

Expected output should be shown.

### Test Results

PASS