

**CS 1632 Software Quality Assurance**

**Exercise 1**

Member 1 Name: Matthew Guiddy

Member 2 Name: Griffin McCool

Team Name: Matt and Griffin

1. **Introduction**
   1. **One Base Case**

Provide the IDENTIFIER of a base case that you tested. Explain why this is a base case.

FUN-RESULTS-TEST: This is a base case because it tests a case with expected behavior in the arguments and all options are in the middle of the valid equivalence class (not on the edge). For example, 120 and 30 as arguments for iterations and threads respectively are both valid arguments and each thread runs 4 iterations. Thus, there is no edge case in terms of those arguments. Also, the equivalence class for number of arguments that are valid is only 4 arguments, so while it is technically on the edge of this class, it is also the only valid number of arguments making it a base case.

* 1. **One Edge Case**

Provide the IDENTIFIER of an edge case that you tested. Explain why this is an edge case.

TEST-ARGS-NUMBER-FIVE-ARGS: This is an edge case because it tests the behavior when five arguments are passed on the command line. Since four arguments are the anticipated number of arguments, five arguments would be an edge case.

* 1. **One Corner Case**

Provide the IDENTIFIER of a corner case that you tested. Explain why this is a corner case.

FUN-SMALL-NUM-TEST-GREATER-LARGE: This is a corner case because the argument “5000000” for number of iterations is much > 100. This is a requirement of FUN-SMALL-NUM. "5000000” is a number far from any edge, therefore it is a corner case.

1. **Traceability Matrix**

Complete the traceability matrix below. An “X” mark in a cell means that the test on the row verifies the requirement on the column. Note that “TEST-CASE-1” and “TEST-CASE-2” are just example test case identifiers. Please replace with your own test cases.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | FUN-ARGS-NUMBER | FUN-ARGS-INVALID | FUN-DISPLAY-RESULTS | FUN-DISPLAY-ITERATIONS | FUN-SMALL-NUM |
| TEST-ARGS-NUMBER-THREE-ARGS | x |  |  |  |  |
| TEST-ARGS-NUMBER-FIVE-ARGS | x |  |  |  |  |
| TEST-INVALID-ITERATIONS-ARG |  | x |  |  |  |
| TEST-INVALID-THREADS-ARG |  | x |  |  |  |
| FUN-RESULTS-TEST | x |  | x | x | x |
| FUN-SMALL-NUM-TEST-GREATER-LARGE | x |  | x | x | x |
| FUN-SMALL-NUM-TEST-LESS |  |  |  |  | x |
| FUN-SMALL-NUM-TEST-EXACT |  |  | x | x | x |
| FUN-SMALL-NUM-TEST-GREATER-SMALL |  |  | x | x | x |
| ITERATIONS-TEST-OVERFLOW |  |  | x | x | x |

1. **Test Cases**

Test case 1:

IDENTIFIER: TEST-ARGS-NUMBER-THREE-ARGS

TEST CASE:

* When three arguments are passed in the command line, the system displays the usage information for the program and halts.

PRECONDITIONS:

* Executing the statement “java –version” in the command line results in the system displaying Java Runtime Environment 1.8 in the terminal window
* The JAR file GoatGoatcar.jar is in the current directory

EXECUTION STEPS:

* Enter the statement “java -jar GoatGoatCar.jar car goat 1000" into the command prompt window and hit the enter key on your keyboard

POST CONDITIONS:

* The system should display only the usage information for the program in the terminal window and nothing else
* The program should halt, causing the terminal window to display the current directory

Test case 2:

IDENTIFIER: TEST-ARGS-NUMBER-FIVE-ARGS

TEST CASE:

* When five arguments are passed in the command line, the system displays the usage information for the program and halts.

PRECONDITIONS:

* Executing the statement “java –version” in the command line results in the system displaying Java Runtime Environment 1.8 in the terminal window
* The JAR file GoatGoatcar.jar is in the current directory

EXECUTION STEPS:

* Enter the statement “java -jar GoatGoatCar.jar car goat 1000 4 100" into the command prompt window and hit the enter key on your keyboard

POST CONDITIONS:

* The system should display only the usage information for the program in the terminal window and nothing else
* The program should halt, causing the terminal window to display the current directory

Test case 3:

IDENTIFIER: TEST-INVALID-ITERATIONS-ARG

TEST CASE:

* When the argument for the number of iterations is entered as something other than a positive integer, the system outputs an explanation of why the program cannot be run and the program halts.

PRECONDITIONS:

* Executing the statement “java –version” in the command line results in the system displaying Java Runtime Environment 1.8 in the terminal window
* The JAR file GoatGoatcar.jar is in the current directory

EXECUTION STEPS:

* Enter the statement “java -jar GoatGoatCar.jar car goat abcd 4" into the command prompt window and hit the enter key on your keyboard

POST CONDITIONS:

* The system should display a message in the terminal window explaining that the program cannot be run and why.
* The program should halt, causing the terminal window to display the current directory

Test case 4:

IDENTIFIER: TEST-INVALID-THREADS-ARG

TEST CASE:

* When the argument for the number of threads is entered as something other than a positive integer, the system outputs an explanation of why the program cannot be run and the program halts.

PRECONDITIONS:

* Executing the statement “java –version” in the command line results in the system displaying Java Runtime Environment 1.8 in the terminal window
* The JAR file GoatGoatcar.jar is in the current directory

EXECUTION STEPS:

* Enter the statement “java -jar GoatGoatCar.jar car goat 1000 abcd" into the command prompt window and hit the enter key on your keyboard

POST CONDITIONS:

* The system should display a message in the terminal window explaining that the program cannot be run and why.
* The program should halt, causing the terminal window to display the current directory

Test case 5:

IDENTIFIER: FUN-RESULTS-TEST

TEST CASE:

* When valid arguments are passed, the system shall display the results of Monty Hall simulation to the user, using percentages with up to three places after the decimal, and then stop execution. This display shall print out the passed-in String versions of the "good" and "bad" options as defined in the arguments. The system shall also display the number of iterations executed by each thread, where there are as many threads as specified in the arguments.

PRECONDITIONS:

* Executing the statement “java –version” in the command line results in the system displaying Java Runtime Environment 1.8 in the terminal window
* The JAR file GoatGoatcar.jar is in the current directory

EXECUTION STEPS:

* Enter the statement “java -jar GoatGoatCar.jar car goat 120 30" into the command prompt window and hit the enter key on your keyboard

POST CONDITIONS:

* The system outputs 30 threads with 4 iterations each and displays the result of the Monty Hall simulation using “car” as the “good” option and “goat” as the “bad” option with accompanying percentages rounded up to 3 decimals places.

Test case 6:

IDENTIFIER: FUN-SMALL-NUM-TEST-GREATER-LARGE

TEST CASE:

* When a very large number greater than 100 is passed as an argument for iterations, the system does not issue a warning.

PRECONDITIONS:

* Executing the statement “java –version” in the command line results in the system displaying Java Runtime Environment 1.8 in the terminal window
* The JAR file GoatGoatcar.jar is in the current directory

EXECUTION STEPS:

* Enter the statement “java -jar GoatGoatCar.jar car goat 2147483647 1" into the command prompt window and hit the enter key on your keyboard

POST CONDITIONS:

* The system does not issue a warning. Then, the system outputs 1 thread with 2147483647 iterations each and displays the result of the Monty Hall simulation using “car” as the “good” option and “goat” as the “bad” option with accompanying percentages rounded up to 3 decimals places.

Test case 7:

IDENTIFIER: FUN-SMALL-NUM-TEST-LESS

TEST CASE:

* When a number less than 100 is passed as an argument for iterations, the system issues a warning.

PRECONDITIONS:

* Executing the statement “java –version” in the command line results in the system displaying Java Runtime Environment 1.8 in the terminal window
* The JAR file GoatGoatcar.jar is in the current directory

EXECUTION STEPS:

* Enter the statement “java -jar GoatGoatCar.jar car goat 80 4" into the command prompt window and hit the enter key on your keyboard

POST CONDITIONS:

* The system issues a warning and asks the user if they wish to continue

Test case 8:

IDENTIFIER: FUN-SMALL-NUM-TEST-EXACT

TEST CASE:

* When exactly 100 is passed as an argument for iterations, the system does not issue a warning.

PRECONDITIONS:

* Executing the statement “java –version” in the command line results in the system displaying Java Runtime Environment 1.8 in the terminal window
* The JAR file GoatGoatcar.jar is in the current directory

EXECUTION STEPS:

* Enter the statement “java -jar GoatGoatCar.jar car goat 100 4" into the command prompt window and hit the enter key on your keyboard

POST CONDITIONS:

* The system does not issue a warning. Then, the system outputs 4 threads with 25 iterations each and displays the result of the Monty Hall simulation using “car” as the “good” option and “goat” as the “bad” option with accompanying percentages rounded up to 3 decimals places.

Test case 9:

IDENTIFIER: FUN-SMALL-NUM-TEST-GREATER-SMALL

TEST CASE:

* When a number slightly greater than 100 is passed as an argument for iterations, the system does not issue a warning.

PRECONDITIONS:

* Executing the statement “java –version” in the command line results in the system displaying Java Runtime Environment 1.8 in the terminal window
* The JAR file GoatGoatcar.jar is in the current directory

EXECUTION STEPS:

* Enter the statement “java -jar GoatGoatCar.jar car goat 101 25" into the command prompt window and hit the enter key on your keyboard

POST CONDITIONS:

* The system does not issue a warning. Then, the system outputs 1 thread with 5 iterations and 24 threads with 4 iterations each and displays the result of the Monty Hall simulation using “car” as the “good” option and “goat” as the “bad” option with accompanying percentages rounded up to 3 decimals places.

Test case 10:

IDENTIFIER: ITERATIONS-TEST-OVERFLOW

TEST CASE:

* When a number greater than 100 that causes integer overflow is passed as an argument for iterations, the system does not issue a warning and continues as normal.

PRECONDITIONS:

* Executing the statement “java –version” in the command line results in the system displaying Java Runtime Environment 1.8 in the terminal window
* The JAR file GoatGoatcar.jar is in the current directory

EXECUTION STEPS:

* Enter the statement “java -jar GoatGoatCar.jar car goat 2147483648 4" into the command prompt window and hit the enter key on your keyboard

POST CONDITIONS:

* The system does not issue a warning. Then, the system outputs 4 threads executing 536870912 iterations each and displays the result of the Monty Hall simulation using “car” as the “good” option and “goat” as the “bad” option with accompanying percentages rounded up to 3 decimals places.

1. **Link to GitHub issues**

Please paste here the URL to your GitHub classroom repository issues page where I can find the three open issues.

<https://github.com/CS1632/exercise-1-test-plans-matt-and-griffin/issues>