



CS 1632 Software Quality Assurance

Exercise 1

Member 1 Name:

Tyler Comisky

Member 2 Name:

Keegan Fouse

1. Traceability Matrix

FUN-ARGS-NUMBER: FUN-THREE-ARGS, FUN-FIVE-ARGS

FUN-ARGS-INVALID: FUN-ARGS-NEGATIVE-INT, FUN-ARGS-NUMBER-TIMES-LARGER-THAN-INT,
FUN-ARGS-NUMBER-THREADS-LARGER-THAN-INT

FUN-DISPLAY-RESULTS: FUN-ARGS-NUMBER-CORRECT

FUN-DISPLAY-ITERATIONS: FUN-ARGS-NUMBER-CORRECT

FUN-SMALL-NUM: FUN-SMALL-NUM-EDGE, FUN-ARGS-NUMBER-LESS

2. Test Cases

IDENTIFIER: FUN-ARGS-NUMBER-LESS

TEST CASE: Test the program with argument num_tries <100

PRECONDITIONS: Arguments "good_choice", "bad_choice" and "num_threads" are valid arguments. JRE version 1.8.0.231.

EXECUTION STEPS:

1. Run the GoatGoatCar program as follows: ">java -jar GoatGoatCar.jar car goat 99 10"
2. Press enter to run the program
3. Observe the behavior of the program

POSTCONDITIONS: Before execution the program will notify you that, "Recommended minimum number of times is 100. Continue? [y/n]"

IDENTIFIER: FUN-ARGS-NUMBER-TIMES-LARGER-THAN-INT

TEST CASE: Test the program with the argument num_tries larger than an integer

PRECONDITIONS: Arguments "good_choice", "bad_choice" and "num_threads" are valid arguments. JRE version 1.8.0.231.

EXECUTION STEPS:

1. Run the GoatGoatCar program as follows: ">java -jar GoatGoatCar.jar car goat 10000000000 10"
2. Press enter to run the program
3. Observe the behavior of the program

POSTCONDITIONS: Immediately after execution the program will tell the user "<num_times> must be an integer greater than 0."

IDENTIFIER: FUN-ARGS-NUM-THREADS-LARGER-THAN-INT

TEST CASE: Test the program with the argument num_threads larger than an integer

PRECONDITIONS: Arguments "good_choice", "bad_choice" and "num_times" are valid arguments. JRE version 1.8.0.231.

EXECUTION STEPS:

1. Run the GoatGoatCar program as follows: ">java -jar GoatGoatCar.jar car goat 1000 100000000000"
2. Press enter to run the program
3. Observe the behavior of the program

POSTCONDITIONS: Immediately after execution the program will tell the user "<num_threads> must be an integer greater than 0."

IDENTIFIER: FUN-ARGS-NUMBER-CORRECT

TEST CASE: Test the program with argument num_tries >100 and a valid number of threads

PRECONDITIONS: Arguments "good_choice", "bad_choice" and "num_times" "num_threads" are valid

arguments. JRE version 1.8.0.231.

EXECUTION STEPS:

1. Run the GoatGoatCar program as follows: ">java -jar GoatGoatCar.jar car goat 101 10"
2. Press enter to run the program
3. Observe the behavior of the program

POSTCONDITIONS: The program will execute as expected and provide you with the viable output.

IDENTIFIER: FUN-ARGS-NEGATIVE-INT

TEST CASE: Test the program with num_tries and num_threads as negative numbers.

PRECONDITIONS: Arguments "good_choice", "bad_choice" are valid arguments. JRE version 1.8.0.231.

EXECUTION STEPS:

1. Run the GoatGoatCar program as follows: "java -jar GoatGoatCar.jar car goat -1 -1".
2. Press enter to run the program.
3. Observe the program's behavior.

POSTCONDITIONS: Immediately after execution the program will tell the user "<num_times> must be an integer greater than 0. <num_threads> must be an integer greater than 0".

IDENTIFIER: FUN-SMALL-NUM-EDGE

TEST CASE: Test the program with the num_tries argument as 100.

PRECONDITIONS: Arguments "good_choice", "bad_choice", and "num_threads" are all valid arguments. JRE version 1.8.0.231.

EXECUTION STEPS:

1. Run the GoatGoatCar program as follows: "java -jar GoatGoatCar.jar car goat 100 4".
2. Press enter to run the program.
3. Observe the program's behavior.

POSTCONDITIONS: The program runs to completion, without any warnings.

IDENTIFIER: FUN-THREE-ARGS

TEST CASE: Test the program by only giving three arguments.

PRECONDITIONS: JRE version 1.8.0.231

EXECUTION STEPS:

1. Run the GoatGoatCar program as follows: "java -jar GoatGoatCar.jar car goat 100".
2. Press enter to run the program.
3. Observe the program's behavior.

POSTCONDITIONS: Immediately after execution the program will tell the user "Usage: java -jar GoatGoatCar.jar <good_choice> <bad_choice> <num_times> <num_threads>".

IDENTIFIER: FUN-FIVE-ARGS

TEST CASE: Test the program by giving five arguments.

PRECONDITIONS: JRE version 1.8.0.231

EXECUTION STEPS:

1. Run the GoatGoatCar program as follows: "java -jar GoatGoatCar.jar car goat 100 100 100".
2. Press enter to run the program.
3. Observe the program's behavior.

POSTCONDITIONS: Immediately after execution the program will tell the user "Usage: java -jar GoatGoatCar.jar <good_choice> <bad_choice> <num_times> <num_threads>".

3. Defects

IDENTIFIER: BUG-NUM-TRIES-100-USAGE-WARNING

SUMMARY: When you input 100 for the num_times argument, it warns you that you should use a minimum of 100, instead of just letting the program run.

DESCRIPTION: When you provide the program with 100 for the num_times argument, it warns you that you should use a value of at least 100 before running the program. According to the FUN-SMALL-NUM requirement, this warning should only occur when the value is strictly less than 100.

REPRODUCTION STEPS:

Preconditions: Arguments "good_choice", "bad_choice", and "num_threads" are all valid arguments. JRE version 1.8.0.231.

Steps:

1. Run the GoatGoatCar program as follows: "java -jar GoatGoatCar.jar car goat 100 4"
2. Press enter in the terminal window to run the program.
3. Observe the warning that appears.

EXPECTED BEHAVIOR: The program should run without any warnings, as intended.

OBSERVED BEHAVIOR: Before running the simulation, the program warns the user that it should use at least 100 tries when the value is exactly 100.

IDENTIFIER: INVALID-NUM-THREAD-STRING-INPUT

SUMMARY: When inputting a non-integer input for num_threads, the program throws an exception.

DESCRIPTION: When using a non-integer input for num_threads, such as "a" or "1000000000000000", the program throws a NumberFormatException. JRE version 1.8.0.231.

REPRODUCTION STEPS:

Preconditions: Arguments "good_choice", "bad_choice", and "num_times" are all valid arguments. JRE version 1.8.0.231

Steps:

1. Run the GoatGoatCar program in the terminal as follows: "java -jar GoatGoatCar.jar car goat 100 abc"
2. Observe the program's behavior.

EXPECTED BEHAVIOR: The program informs the user of the proper syntax and gracefully exits the program.

OBSERVED BEHAVIOR: A NumberFormatException is thrown, and the program exits.

IDENTIFIER: BUG-PERCENTAGE-IN-CHOICE-INPUT

SUMMARY: When "good_choice" or "bad_choice" contains a '%', the program throws an exception.

DESCRIPTION: When you use a % in the "good_choice" or "bad_choice" input, the program throws an UnknownFormatConversionException. JRE version 1.8.0.231.

REPRODUCTION STEPS:

Preconditions: Arguments "num_times" and "num_threads" are valid arguments.

Steps:

1. Run the GoatGoatCar program in the terminal as follows: "java -jar GoatGoatCar.jar car g%oat

101 10"

2. Observe the program's behavior.

EXPECTED BEHAVIOR: The program runs as intended, showing the number of iterations per thread and the results of the simulation.

OBSERVED BEHAVIOR: Immediately upon execution the program throws a "MissingFormatArgumentException".

IDENTIFIER: BUG-MAX-INT-FOR-THREAD-AND-TRIES

SUMMARY: When you use the maximum value for int in both "num_tries" and "num_threads", an exception is thrown. JRE version 1.8.0.231.

DESCRIPTION: When the value 2147483647 is used for both "num_tries" and "num_threads", an OutOfMemoryError is thrown. For other large values that are slightly less than the max int value, the program might run for some time and then throw an OutOfMemoryError.

REPRODUCTION STEPS:

Preconditions: Arguments "good_choice" and "bad_choice" are valid arguments. JRE version 1.8.0.231.

Steps:

1. Run the GoatGoatCar program in the terminal as follows: "java -jar GoatGoatCar.jar car goat 1000000000 1000000000"

2. Observe the program's behavior.

EXPECTED BEHAVIOR: The program runs as intended, showing the number of iterations per thread and the results of the simulation.

OBSERVED BEHAVIOR: The program crashes, either immediately after running the program or while the program is printing the threads.

NOTES: This bug could likely be fixed by using a larger heap size, but an average computer will not have more than 16GB of RAM. Also, the max number that you can use will vary depending on what your heap size is, but generally using two numbers that are both greater than 1 billion will cause an issue.