CS 1632 Software Quality Assurance

Exercise 1

Member 1 Name: David Simpkins (dbs29@pitt.edu)

1. Introduction
    a. Division of Work
        - I worked alone on this exercise.
    b. One Base Case
        - FUN-CORRECT-THREADS-VALUE
            1. This is a base case because it provides a correct value in each argument that each fall within our boundary constraints for our program. This produces expected and realistic values.
    c. One Edge Case
        - FUN-DUPLICATE-INPUT
            1. This is an edge case because it is an unexpected use case in which both choices for arguments 1 and 2 are both the same and thus should allow our program to run but the results will not be indicative of what our program is supposed to actually display and compute.
    d. One Corner Case
        - FUN-EXTRA-ARGS
            1. This is a corner case because 5 arguments is outside the range of our expected or allowed size of our argument array that we are reading from. This should cause an issue in execution.

2. Traceability Matrix

	FUN-ARGS-NUMBER	FUN-ARGS-INVALID	FUN-DISPLAY-RESULTS	FUN-DISPLAY-ITERATIONS	FUN-SMALL-NUM
FUN-CORRECT-THREADS-VALUE			X	X	
FUN-NO-ARGS	X	X			
FUN-WRONG-INPUT-TYPE		X			
FUN-DUPLICATE-INPUT		X			
FUN-EXTRA-ARGS	X				
FUN-NO-ITERATIONS			X	X	X

3. TEST CASES

IDENTIFIER: FUN-CORRECT-THREADS-VALUES

- **Test Case**: Test that the execution of the program properly displays the correct number of values with the correct number of threads.
- **Precondition**: Ensure that GoatGoatCar.jar is in the current directory and Java 8 version "1.8.0_23" is the version on your machine.
- **Execution Steps**:
  1. Input "java -jar GoatGoatCar.jar car goat 10001 4
- **Postconditions**: Given the number of iterations of 10001 the tester should display 1 thread with a value of 2501 and 3 threads with values of 2500.

IDENTIFIER: FUN-NO-ARGS

- **Test Case:** Test that the execution of the program fails on not feeding the command line any inputs.
- **Precondition:** Ensure that Java 8 is installed, and check to see that it is "java 1.8.0_231" is the version installed on your machine. Make sure that GoatGoatCar.jar is in the current directory.
- **Execution Steps:** Input "java -jar GoatGoatCar.jar "
- **Postconditions:** The program should return an error to the user or an exception. Program should not execute due to not having any arguments.

IDENTIFIER: FUN-WRONG-INPUT-TYPE

- **Test Case:** Test that the execution of the program fails upon feeding the wrong data type into one of our arguments.
- Precondition: Ensure that Java 8 is installed, and check to see that it is "java 1.8.0_123" that is the version that is installed on your machine. Make sure GoatGoatCar.jar is in the current working directory.
- **Execution Steps:** input "java -jar GoatGoatCar.jar 20 car 10001 4"
- Postconditions: The program should return an error due to incorrect input type.

IDENTIFIER: FUN-EXTRA-ARGS

- **Test Case:** Test that the execution of the program can handle more than 4 arguments. Test that it will still execute the first 4 as the only valid inputs.

- Precondition: Ensure that Java 8 is installed, and check to see that it is "java 1.8.0_123" that is the version that is installed on your machine.
- Execution Steps: input "java -jar GoatGoatCar.jar goat car 10001 4 hello"
- Postconditions: the program should still function as normal and ignore any extra number of arguments.

IDENTIFIER: FUN-NO-ITERATIONS

- Test Case: Test that without giving a number of iterations (or a 3rd argument that is an integer) the program returns an error message correcting the user of their mistake and outlining the requirements for input.
- Preconditions: Ensure that Java 8 is installed, and check to see that it is "java 1.8.0_123" that is the version that is installed on your machine. Make sure GoatGoatCar.jar is in the current working directory.
- Execution steps: input "java -jar GoatGoatCar.jar goat car hello 4"
- Postconditions: the program should return an error message to the user if it is missing a third argument or the third argument is not an integer data type.

DEFECTS

1. Duplicate choices:
    a. Summary: No error message returned to user if the good choice and bad choice are duplicated.
    b. Description: If the user inputs the same argument for the good and bad choice the program still runs and returns a value that is not correct for what the game is trying to accomplish.
    c. Reproduction Steps:
        i. Preconditions: Ensure that Java 8 is installed, and check to see that it is "java 1.8.0_123" that is the version that is installed on your machine. Make sure GoatGoatCar.jar is in the current working directory.
        ii. Input "java -jar GoatGoatCar.jar goat goat 10001 4
    d. Expected Behavior: Error message should be returned to the user specifying that input for args 1 and 2 should be different so the game can be played correctly.
    e. Observed Behavior: Upon execution the program runs normally with no errors or pause in execution. The computations are made but are now incorrect for the purpose and requirements of the game.
    f. Impact: Duplicate inputs for args 1 and 2 will lead the user to actually initialize a different game being played with different rules than was intended in the requirements.
    g. Severity:
        i. Minor severity.
        ii. Will only occur occasionally.
        iii. There is a workaround to the issue and that involves added conditional statements within the evaluation of the input arguments.
    h. Notes:

2. Thread Value is Greater than Iterations
    a. Summary: No error message upon inputting an integer for argument 4 that is greater than argument 3.
    b. Description: If the user inputs a larger number in arg 4 than in arg 3 the program will still execute and display more threads than needed.
    c. Reproduction Steps:
        i. Preconditions: Ensure that Java 8 is installed, and check to see that it is "java 1.8.0_123" that is the version that is installed on your machine. Make sure GoatGoatCar.jar is in the current working directory.
        ii. Input "java -jar GoatGoatCar.jar goat car 100 120"
    d. Expected Behavior: Program should only display the threads that are actually in use.
    e. Observed Behavior: The program will list all threads even those that are not in use before returning the calculations to the user.
    f. Impact: With very large values for arguments 3 and 4 and depending on the machine the program may timeout.

g. Severity
    i. Minor severity
    ii. Will only occur occasionally
    iii. Workaround includes more conditionals in the input reading.