Assignment - ASE

Component - 1

Pravesh Pansari

2020

Contents

[Simple Programming Environment 2](#_Toc58873804)

[Interface: 2](#_Toc58873805)

[Functions: 3](#_Toc58873806)

[Commands: 4](#_Toc58873807)

[Program Design 5](#_Toc58873808)

[UML Class Diagram 5](#_Toc58873809)

[Version Control – GitHub 6](#_Toc58873810)

[Repository: 6](#_Toc58873811)

[Commits: 7](#_Toc58873812)

[Test Cases 9](#_Toc58873813)

[Summary: 9](#_Toc58873814)

[All tests pass: 9](#_Toc58873815)

[All tests fail: 9](#_Toc58873816)

[Results: 10](#_Toc58873817)

[Shape Factory Test – Rectangle 10](#_Toc58873818)

[Shape Factory Test – Triangle 10](#_Toc58873819)

[Command Test – Move to 11](#_Toc58873820)

[Command Test – Draw To 11](#_Toc58873821)

[Command Test – Reset Command 12](#_Toc58873822)

[Command Test – Pen Command 12](#_Toc58873823)

[Command Test – Create Variable 13](#_Toc58873824)

[Command Test – While 13](#_Toc58873825)

[Command Test – If statement 14](#_Toc58873826)

[Command Test – Create Method 14](#_Toc58873827)

[Command Test – Complex Expression 15](#_Toc58873828)

# Simple Programming Environment

## Interface:

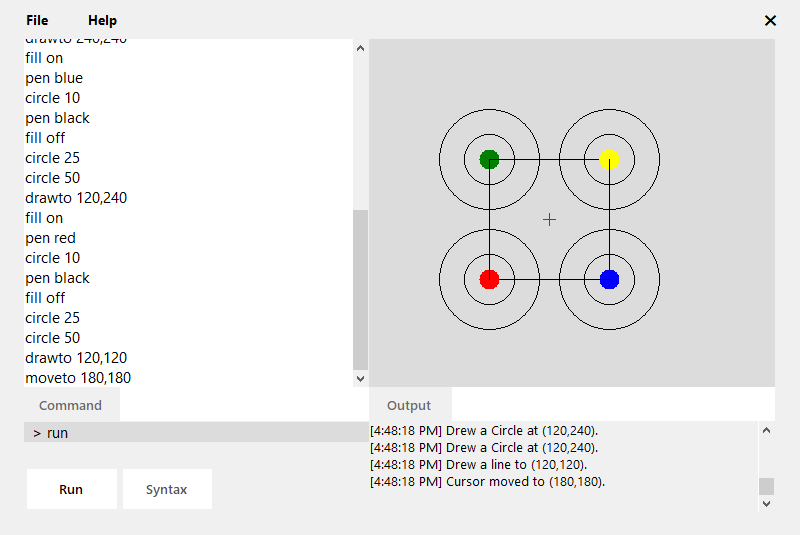


Figure : The GUI of Simple Programming Environment

This program creates a simple programming environment, where commands can be used to manipulate shapes and lines on an artboard. The artboard is the place where all drawings and the cursor are rendered. The menu allows access to different file operations, exit, and the about information. The current position of the cursor on the artboard is represented by the red cross. The command line runs single commands while the code editor can run multiple commands where every command is on a new line. Commands can be executed using the run button. The output box displays the log information and error information for the commands. The syntax button analyzes the code in the code editor and displays any errors it found in the log box.

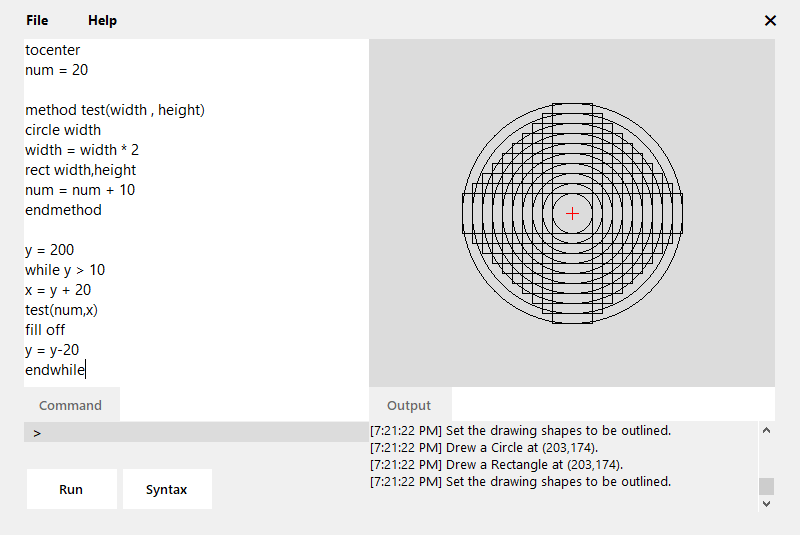


Figure : Making a complex drawing using advanced commands

## Functions:

Four types of shapes can be drawn: rectangle, square, circle and triangle. And a line can be drawn given a destination position. The shapes can be drawn outlined or filled. The color of shapes and lines can be changed. The cursor can be reset to top left, moved to center or any position in the artboard. The board can be cleared, which removes all drawings on it. The file menu allows the user to create new file, save the code, open a file and exit the program. The help menu consists of information about the program.

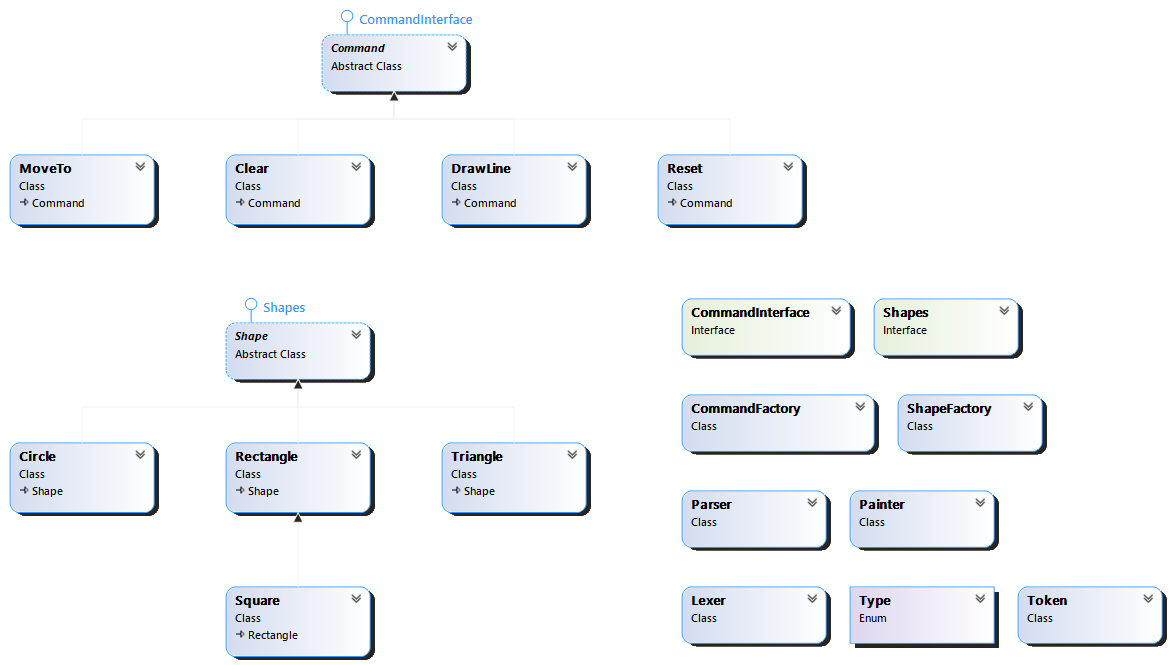
Variables can be declared in this program. They must contain only letters and can be assigned to integers only. Complex operations can be performed in the assignment statement including 5 types of mathematical operations. Commands can be run conditionally using if statements. They can be used as single line or a complete block of code. While loops can be used to redo commands until a stop condition. Methods can be declared and be called. Both parameterized and parameter less methods can be implemented. Then they can be called by passing appropriate parameters.

## Commands:

* **drawto** int x, int y – Draws a line from current position to (x, y)
* **moveto** int x, int y – Moves the cursor to (x, y)
* **pen** string color – Changes the Pen color to color
* **fill** string input – Sets the fill of the shapes (on or off)
* **rect** int width, int height – Draws a rectangle of width and height
* **square** int size – Draws a square of length size
* **circle** int radius – Draws a circle of radius
* **triangle** int base, int height – Draws a triangle with base and height
* **clear** – Clears the drawing area
* **reset** – Reset the circle position to (0, 0)
* **tocenter** – Moves the cursor to center
* string test **=** int value – Creates a variable named test and assigns value to it
* **if** var **op** int x – Compares var against x using comparison operator op
* **while** var **op** int x – Loops till the condition var against x is true using op
* **method** example (param) – Creates a function example with parameter param

# Program Design

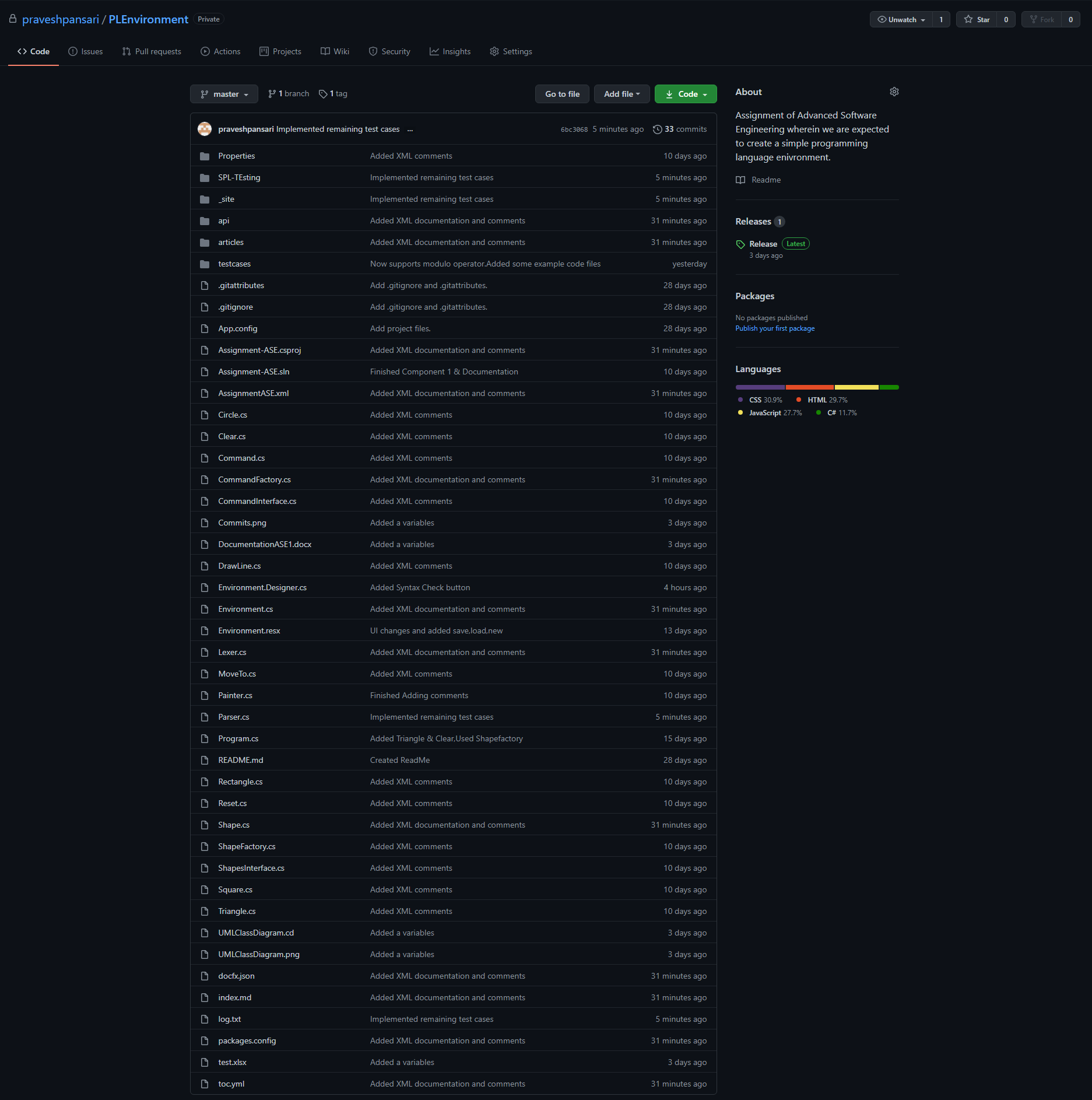
## UML Class Diagram



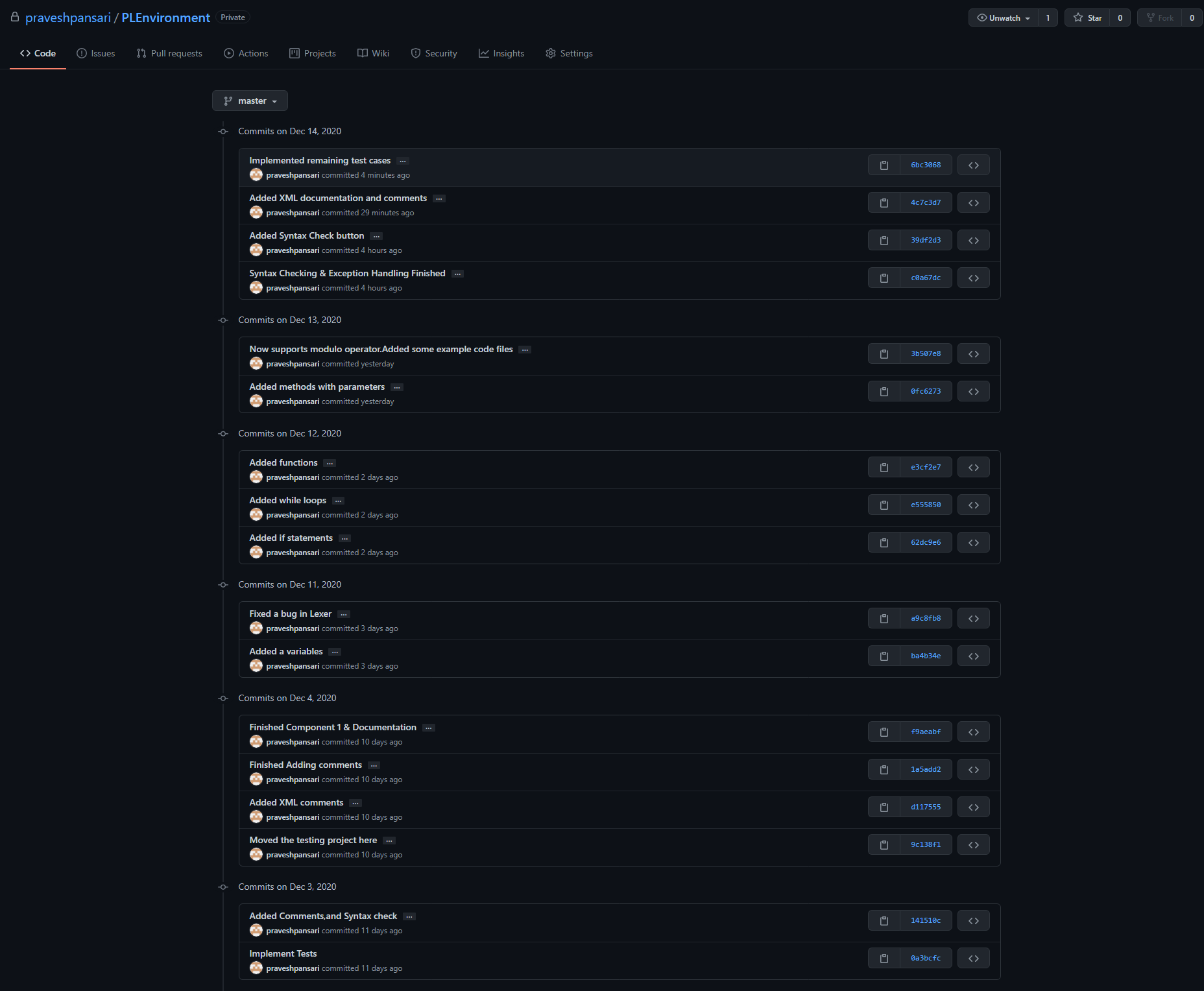
# Version Control – GitHub

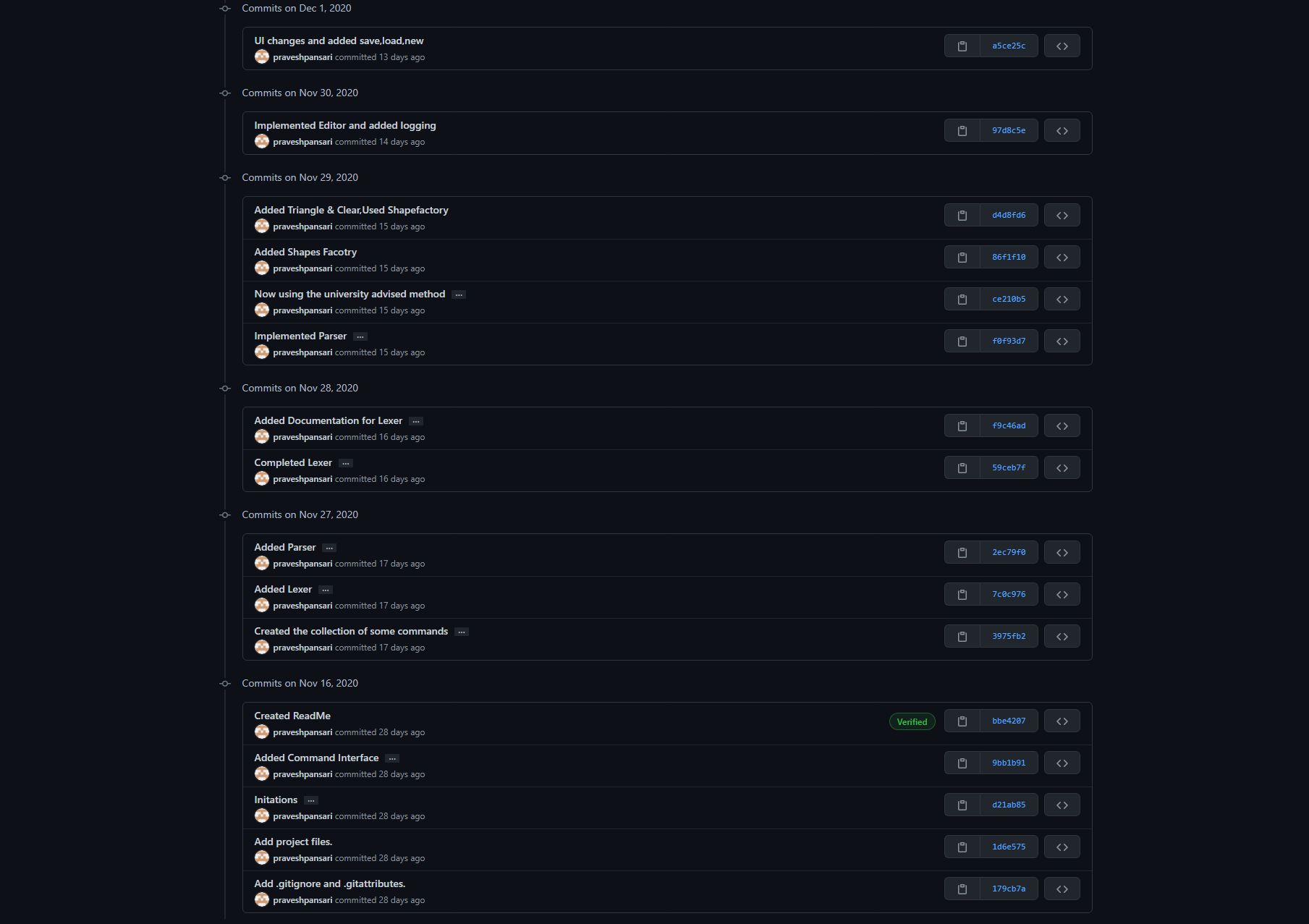
## Repository:

#### Link: <https://github.com/praveshpansari/PLEnvironment>



## Commits:

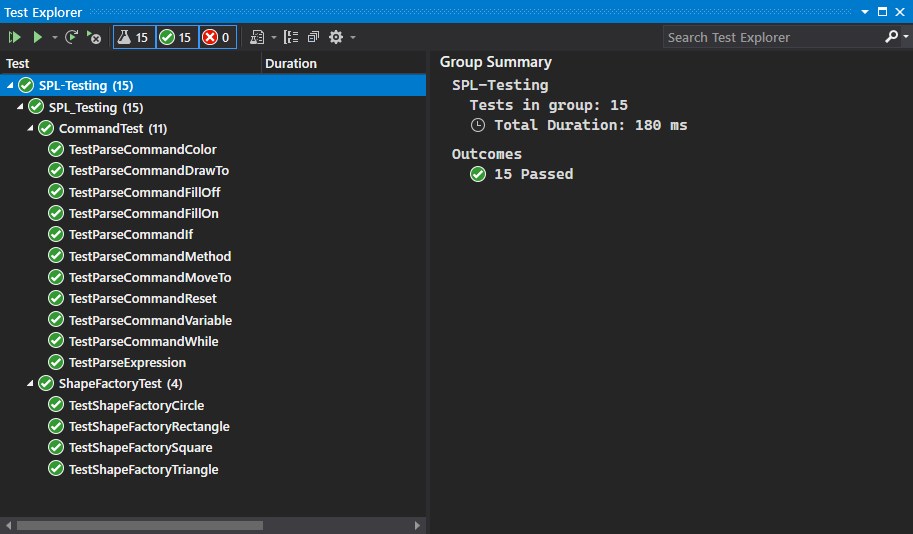




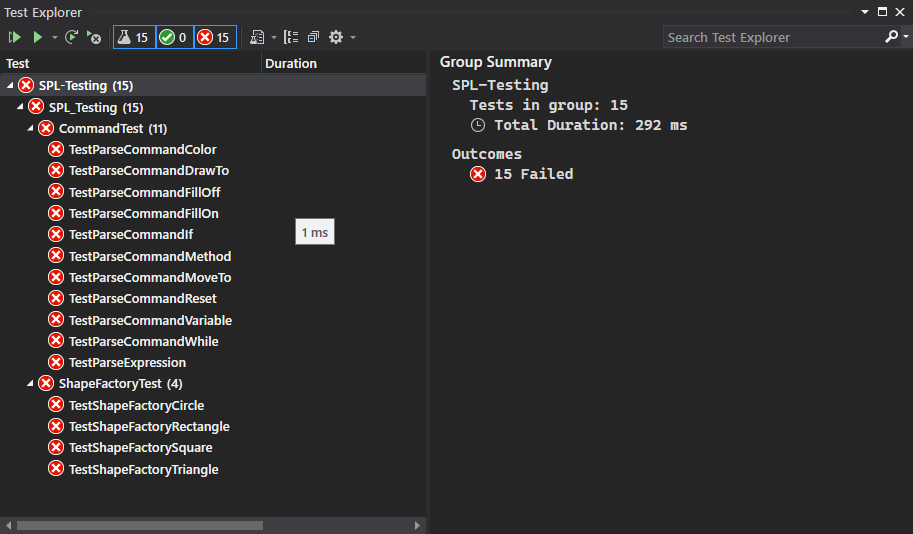
# Test Cases

## Summary:

### All tests pass:

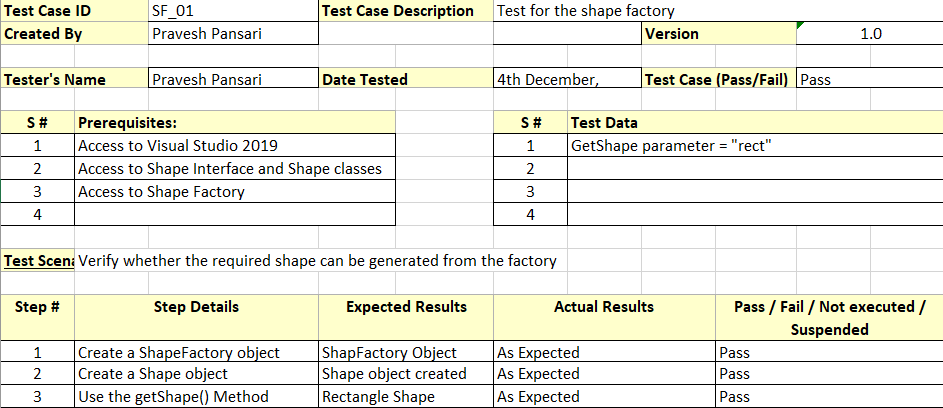


### All tests fail:

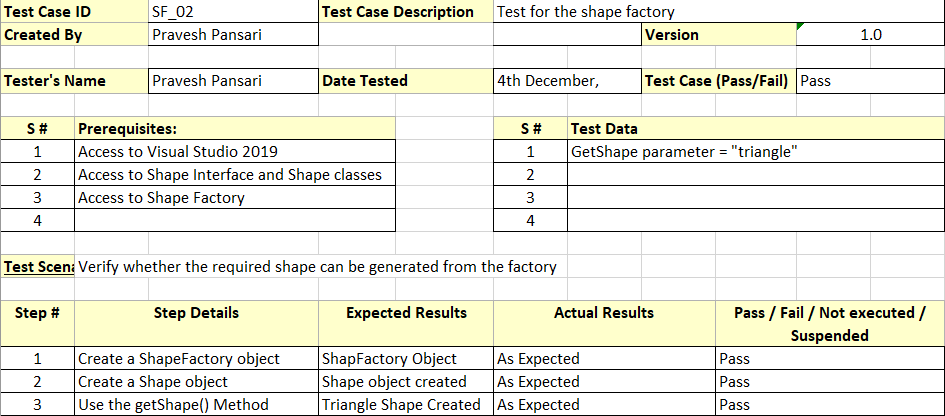


## Results:

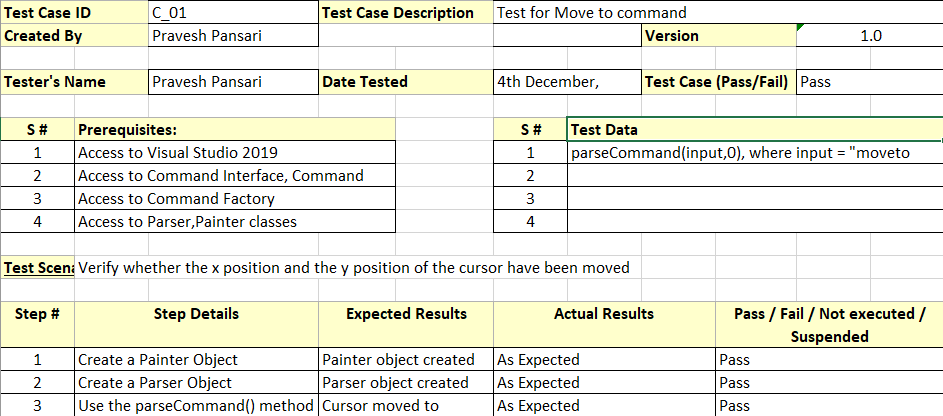
### Shape Factory Test – Rectangle



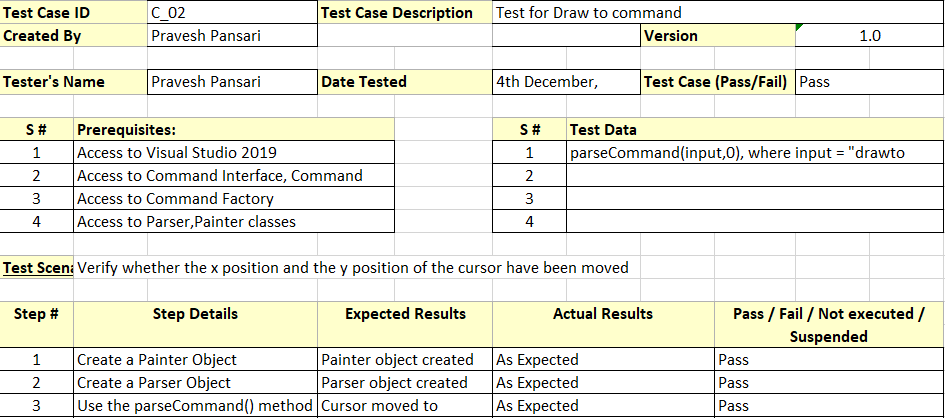
### Shape Factory Test – Triangle



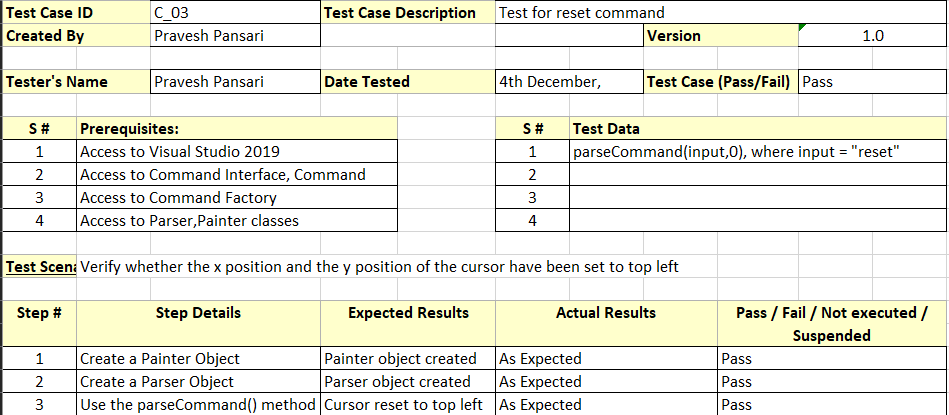
### Command Test – Move to



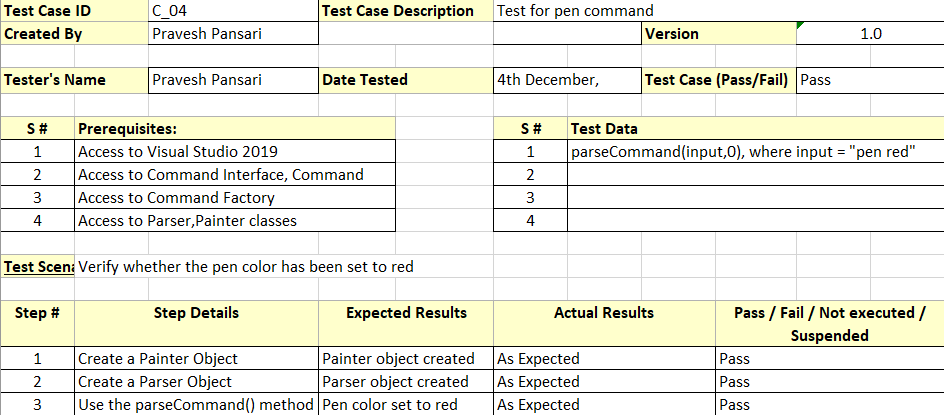
### Command Test – Draw To



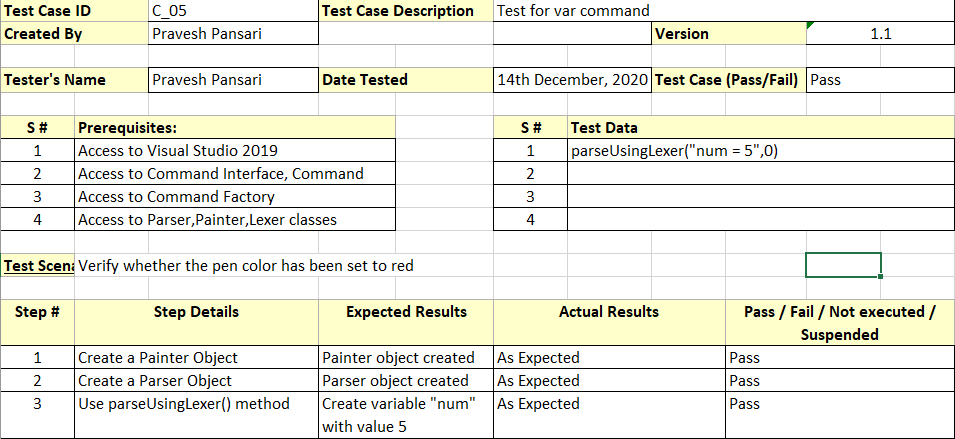
### Command Test – Reset Command



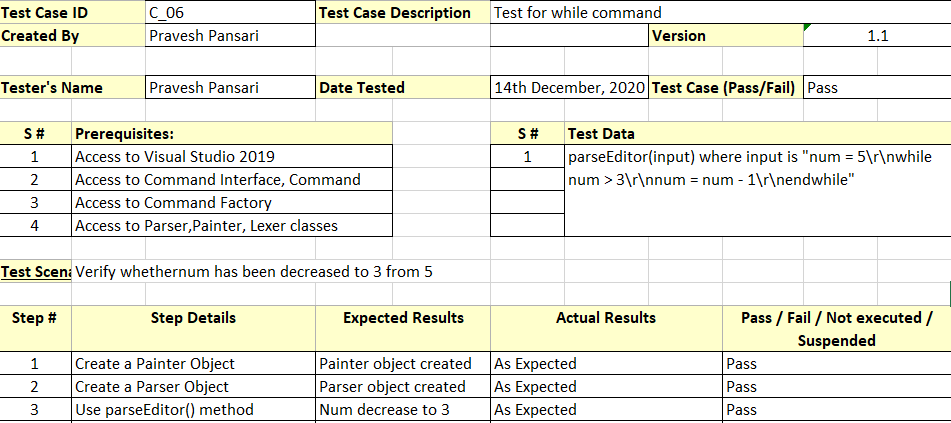
### Command Test – Pen Command



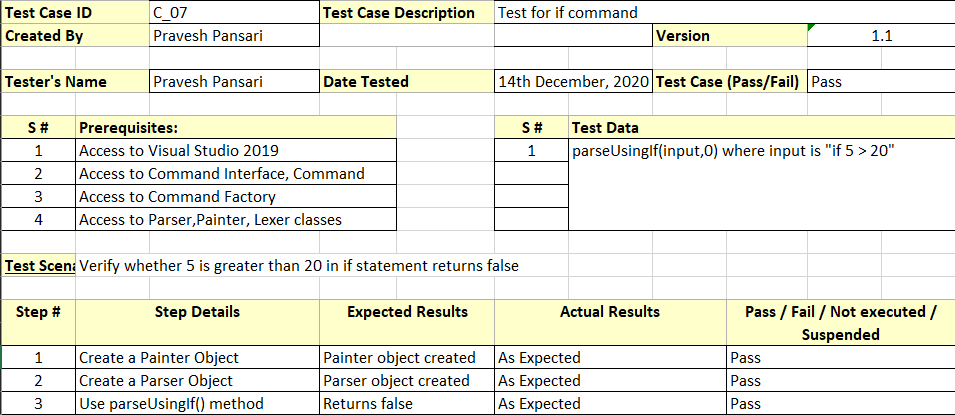
### Command Test – Create Variable



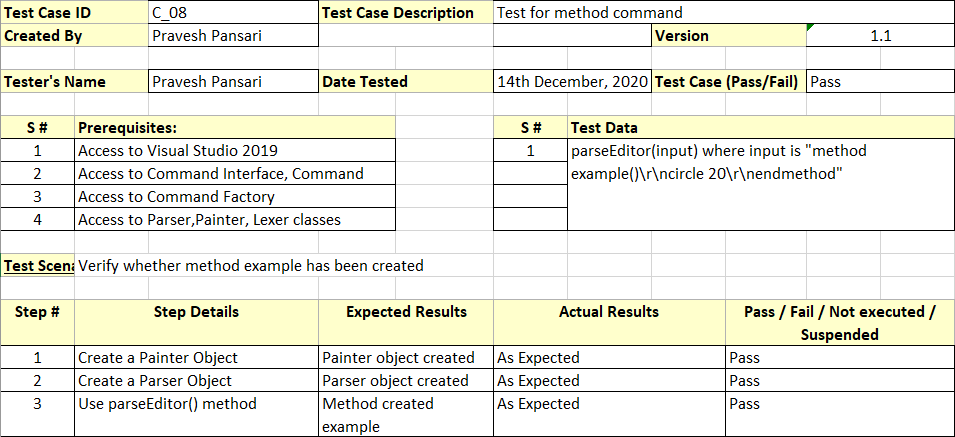
### Command Test – While



### Command Test – If statement



### Command Test – Create Method



### Command Test – Complex Expression

