

Advanced Software Engineering

Submitted To Resham Pun

By Rakshya Lama Moktan(L6) C7227218

TABLE OF CONTENT

[1. Version Control 5](#_Toc59725361)

[2. Basic Commands Execution 6](#_Toc59725362)

[a. Circle 6](#_Toc59725363)

[b. Rectangle 6](#_Toc59725364)

[c. Triangle 7](#_Toc59725365)

[d. Pen Command 7](#_Toc59725366)

[e. Fill command 8](#_Toc59725367)

[f. MoveTo Command 8](#_Toc59725368)

[g. DrawTo Command 9](#_Toc59725369)

[h. Polygon Command 9](#_Toc59725370)

[i. Rotate 10](#_Toc59725371)

[j. Flash Command 10](#_Toc59725370)

[3. Save and load 10](#_Toc59725372)

[a. Save commands into text file 10](#_Toc59725373)

[b. Load commands 12](#_Toc59725374)

[4. Command validation check 13](#_Toc59725375)

[a. Valid commands and execution 13](#_Toc59725376)

[b. Invalid Parameter Execution 14](#_Toc59725377)

[c. Invalid Command Execution 14](#_Toc59725378)

[5. If Endif/ Then Command 15](#_Toc59725379)

[a. When the correct code is entered inside if endif 15](#_Toc59725380)

[b. When the correct code is entered inside if then 15](#_Toc59725381)

[c. When the variable is not declared 16](#_Toc59725382)

[d. When condition inside if is not complete 16](#_Toc59725383)

[e. When condition does not match 17](#_Toc59725384)

[f. When if command is not closed 17](#_Toc59725385)

[g. When endif is found before if command 18](#_Toc59725386)

[6. Variable Command 18](#_Toc59725387)

[a. When correct variable is declared. 18](#_Toc59725388)

[b. When the variable is incorrectly declared. 19](#_Toc59725389)

[c. When the same variable is declared again. 19](#_Toc59725390)

[7. Variable Operation Command 20](#_Toc59725391)

[a. When the variable operation is correct. 20](#_Toc59725392)

[b. When the variable is not present 20](#_Toc59725393)

[8. Loop command 21](#_Toc59725394)

[a. When valid loop command is run 21](#_Toc59725395)

[b. When the counter variable is not declared. 21](#_Toc59725396)

[c. When the loop is not closed 22](#_Toc59725397)

[d. When the counter variable is not handled 22](#_Toc59725398)

[9. Method Command 23](#_Toc59725399)

[a. When the method declaration is valid. 23](#_Toc59725400)

[b. When parameter is Numeric 23](#_Toc59725401)

[c. When same method is declared twice 24](#_Toc59725402)

[d. When method call has different parameters than declaration 24](#_Toc59725403)

[e. When the name of method is invalid 25](#_Toc59725404)

[1. Method name starting with number 25](#_Toc59725405)

[2. Method name containing invalid characters 25](#_Toc59725406)

[f. When method syntax is invalid 26](#_Toc59725407)

[g. When method is not closed 26](#_Toc59725408)

[h. When method is closed before or without method 27](#_Toc59725409)

[10. Singleton Design Pattern 27](#_Toc59725410)

[11. Unit Testing 29](#_Toc59725411)

[a. Testing valid drawing commands 29](#_Toc59725412)

[1. Working process of this test 32](#_Toc59725413)

[i. If Wrong Parameter is Passed, test fails. 32](#_Toc59725414)

[ii. If Parameter is Passed Correctly 33](#_Toc59725415)

[2. Screenshots of valid commands test 33](#_Toc59725416)

[i. Run, Clear, Reset 33](#_Toc59725417)

[ii. MoveTo Command 34](#_Toc59725418)

[iii. DrawTo Command 35](#_Toc59725419)

[iv. Fill 36](#_Toc59725420)

[v. Pen 36](#_Toc59725421)

[vi. Rectangle 37](#_Toc59725422)

[vii. Triangle 38](#_Toc59725423)

[viii. Polygon 39](#_Toc59725424)

[ix. Rotate 40](#_Toc59725425)

[x. Circle 41](#_Toc59725426)

[b. If Endif/Then Command 42](#_Toc59725427)

[1. Testing Code 44](#_Toc59725428)

[2. Test Summary 44](#_Toc59725429)

[c. Loop Command 45](#_Toc59725430)

[1. Testing Code 47](#_Toc59725431)

[2. Test Summary 47](#_Toc59725432)

[d. Method Command 48](#_Toc59725433)

[1. Test Code 50](#_Toc59725434)

[2. Test Summary 50](#_Toc59725435)

[e. Variable Command 51](#_Toc59725436)

[1. Test Code 53](#_Toc59725437)

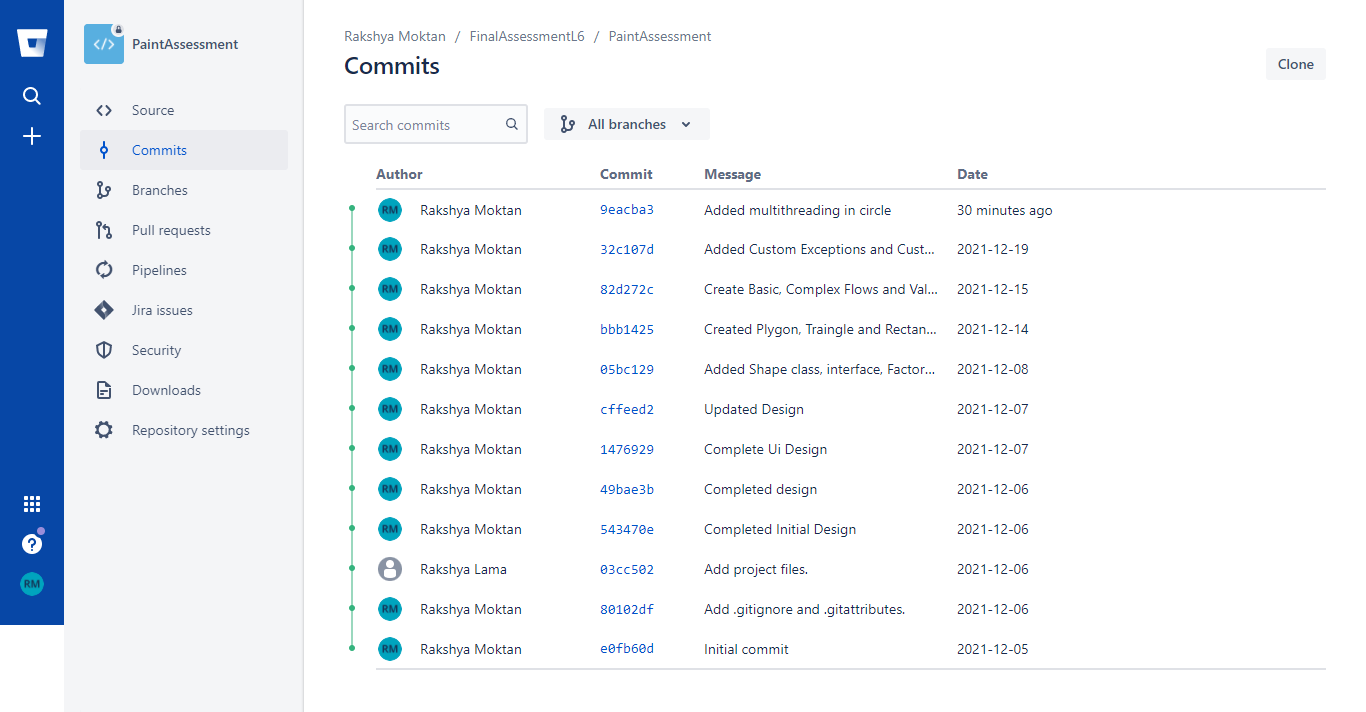
[2. Test Summary 53](#_Toc59725438)

[f. Screenshot of all test passed 54](#_Toc59725439)

[12. Uml Diagram 55](#_Toc59725440)

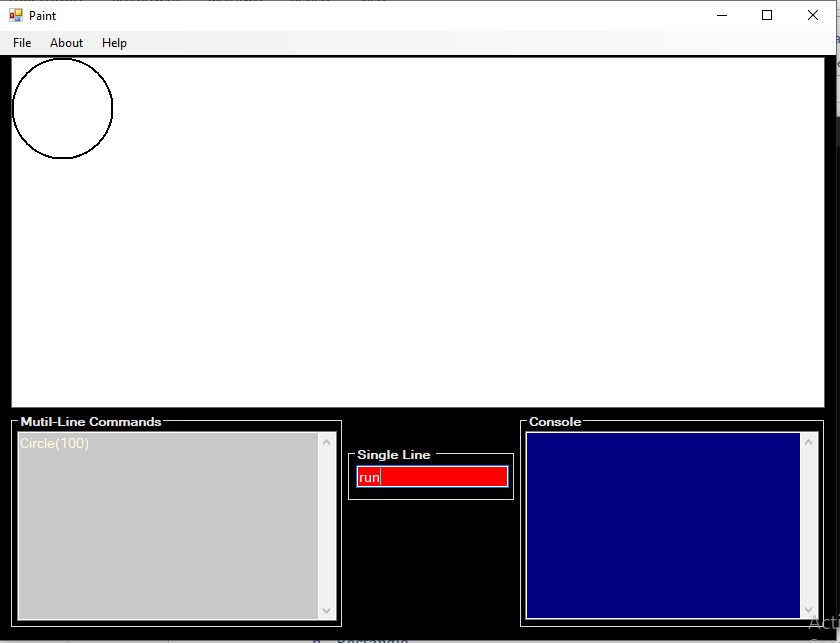
# Version Control

Commits with description in GitHub.



# Basic Commands Execution

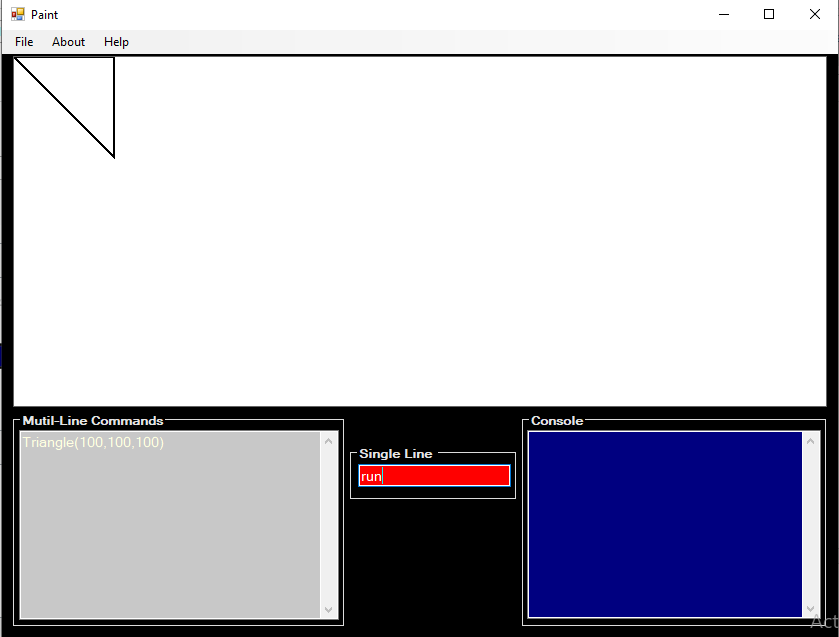
## Circle



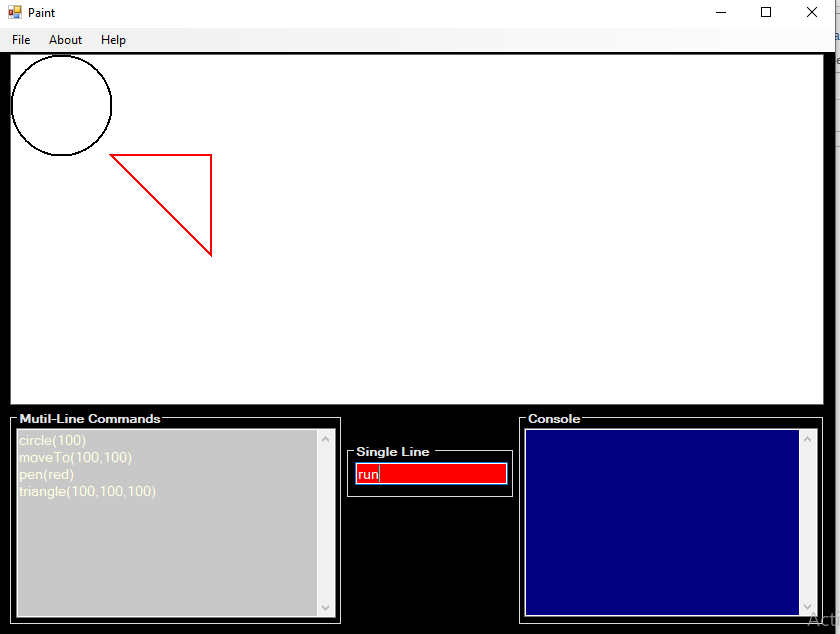
## Rectangle



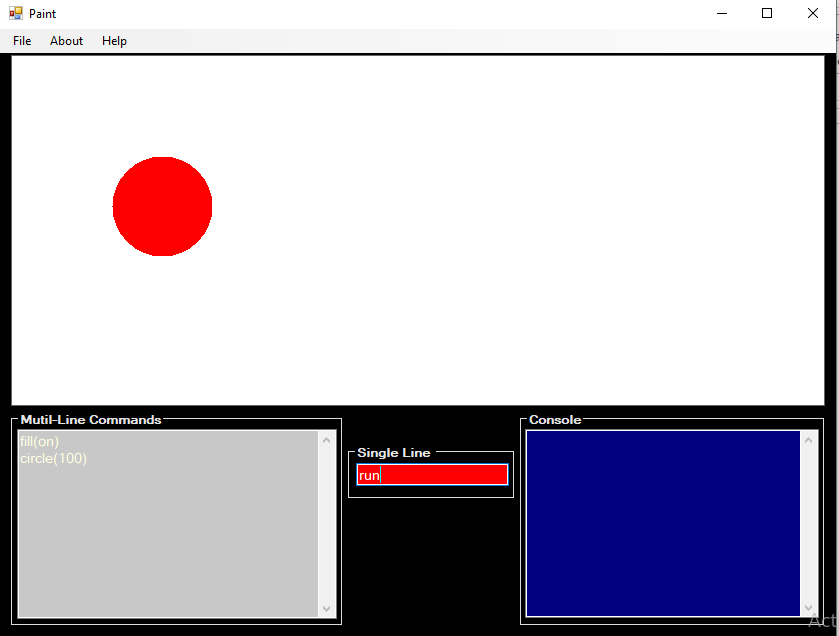
## Triangle



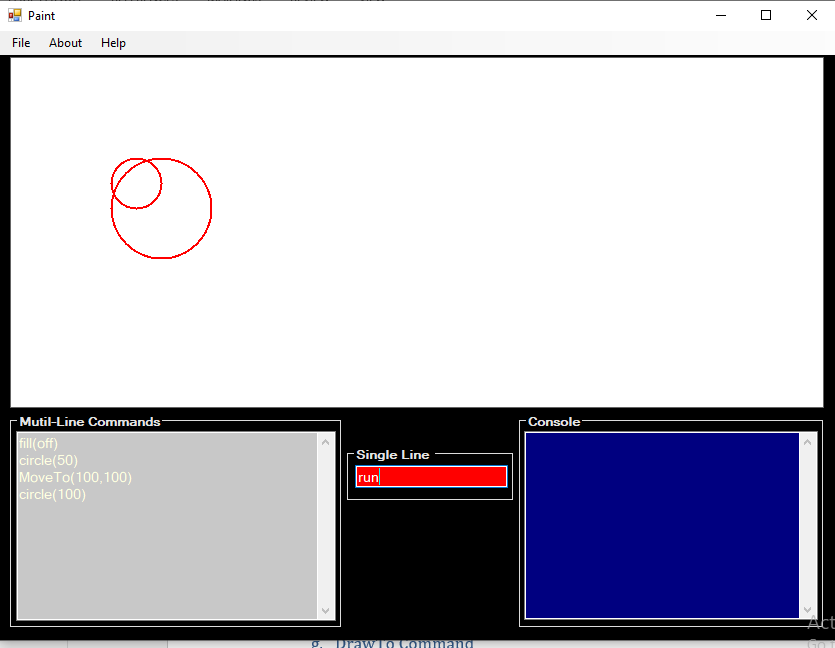
## Pen Command



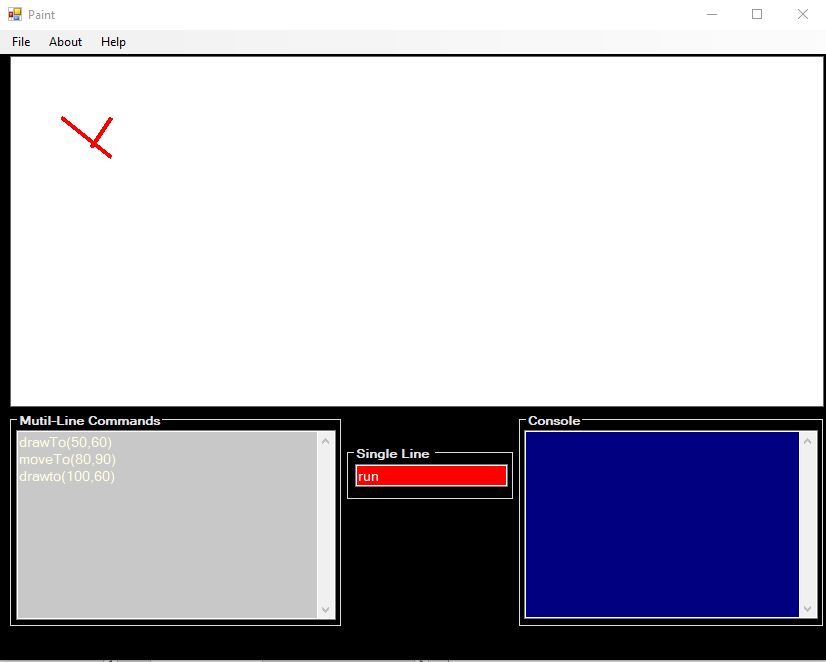
## Fill command



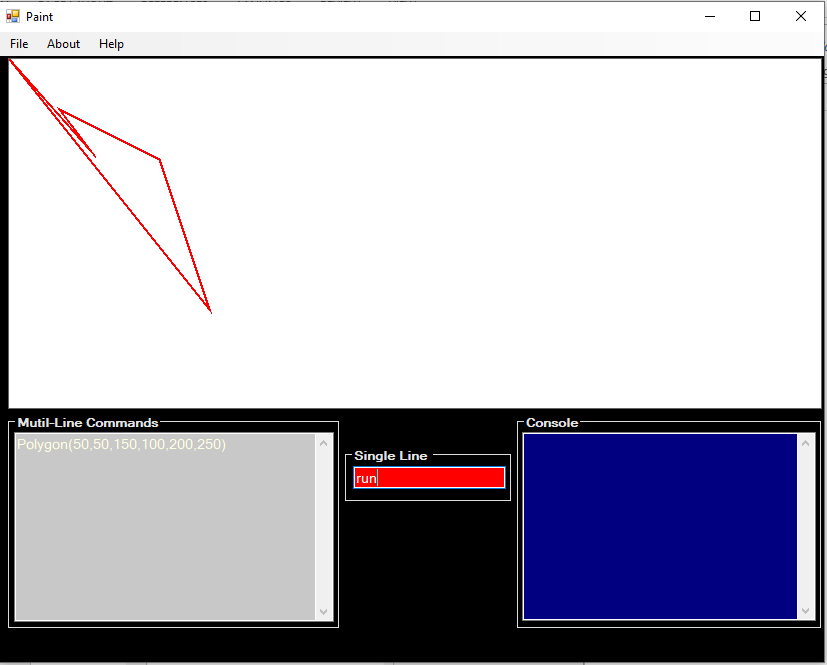
## MoveTo Command



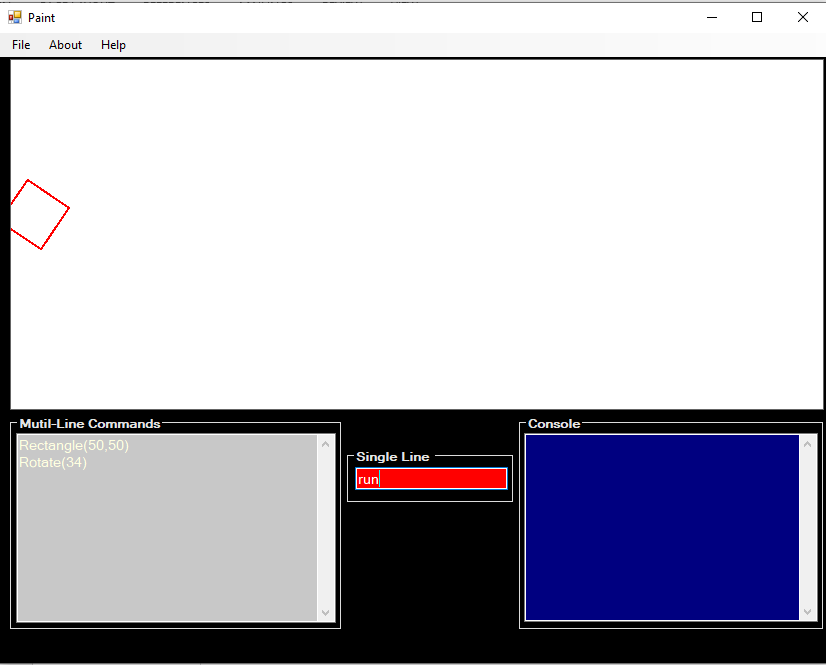
## DrawTo Command



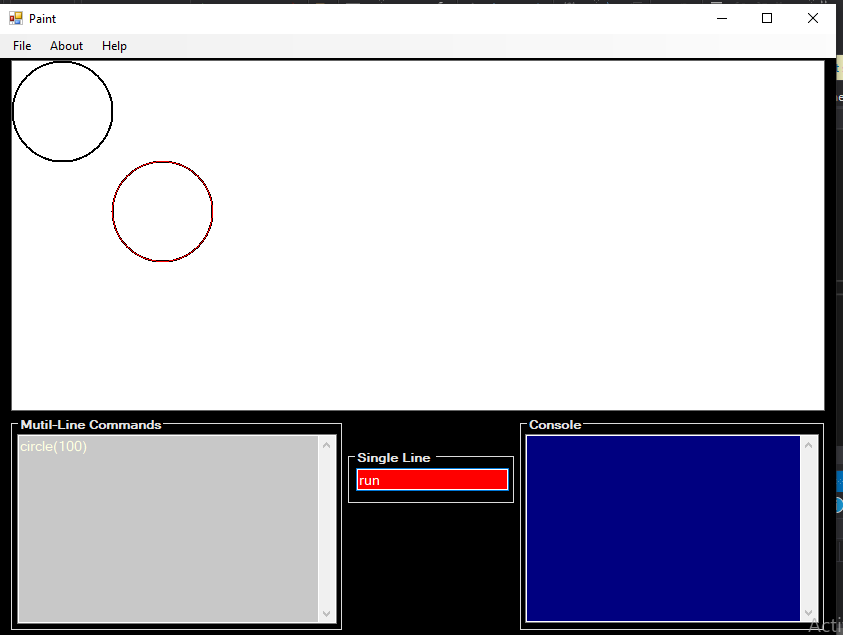
## Polygon Command



## Rotate



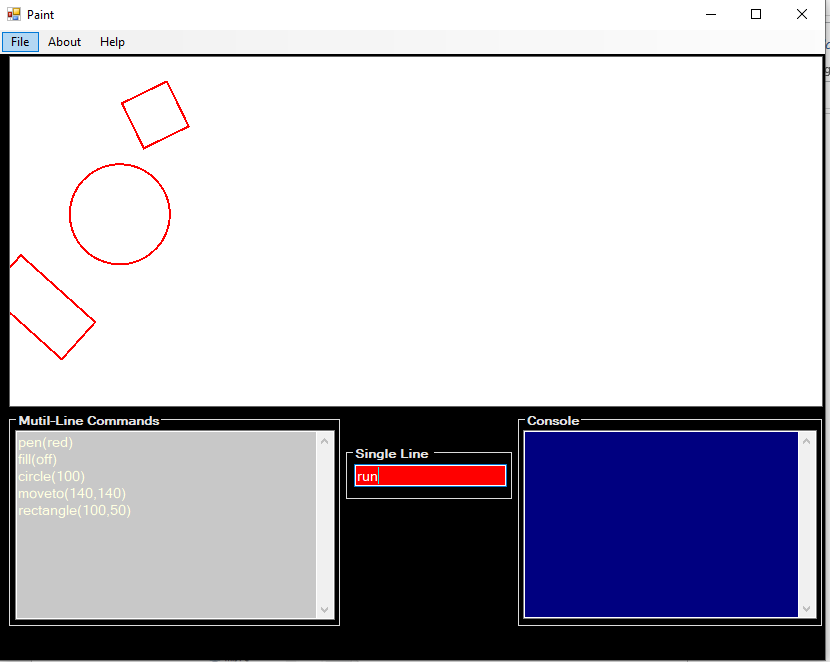
## Flash using Multithreading



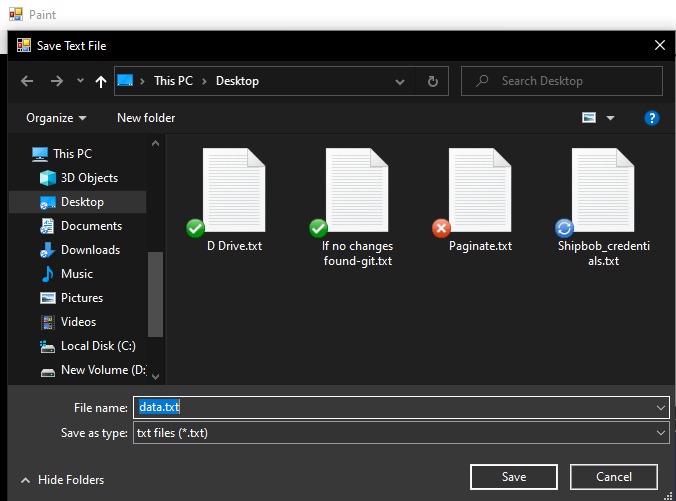
# Save and load

### Save commands into text file

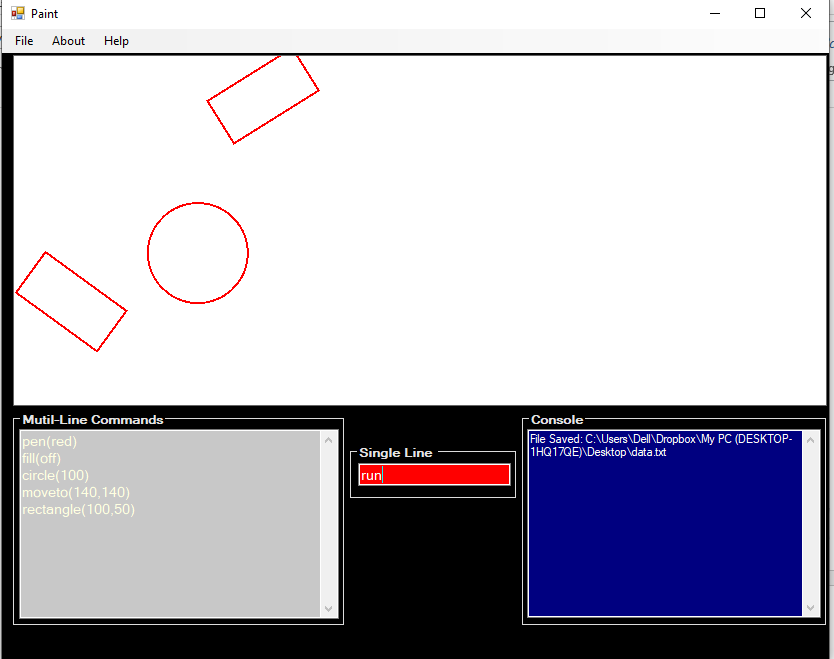
Step 1: Click on save button



Step 2: Save Dialog Box to save file

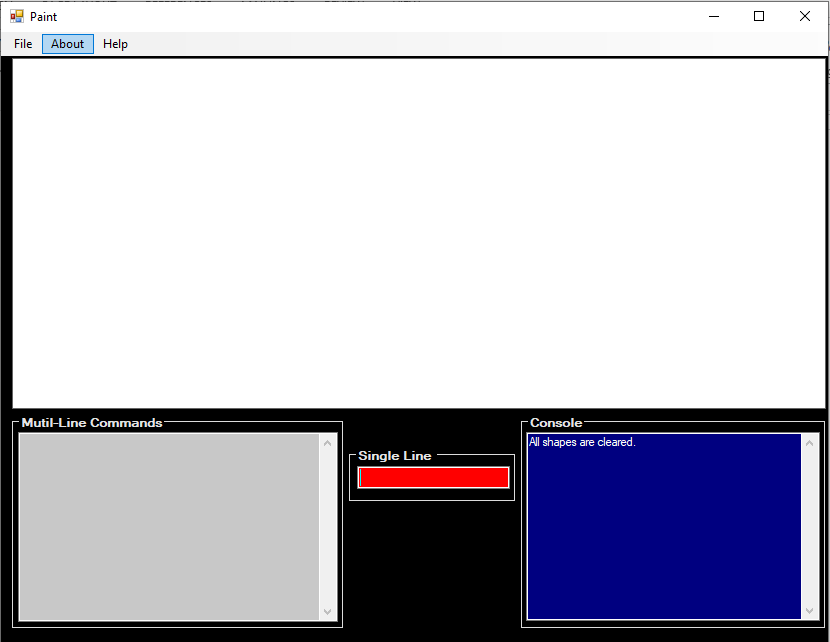


Step 3: Saved successfully.

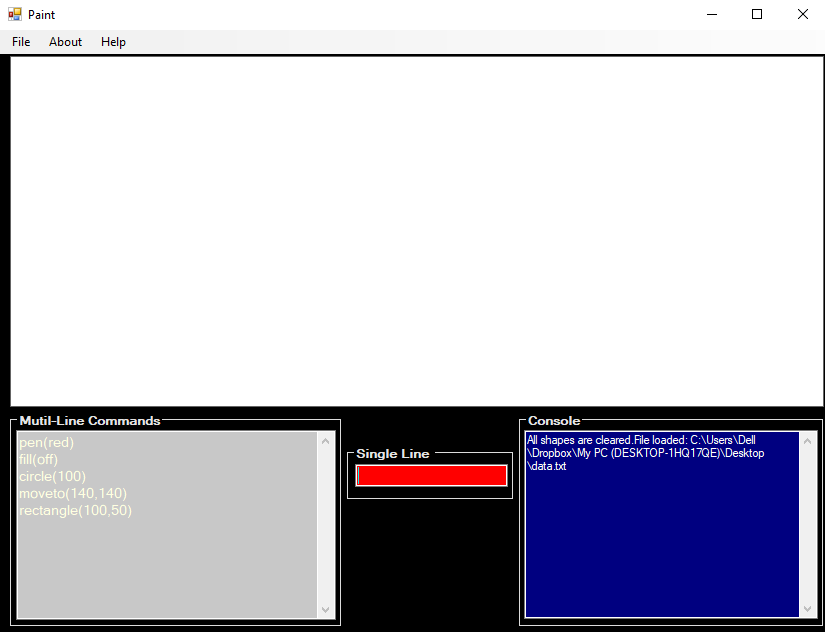
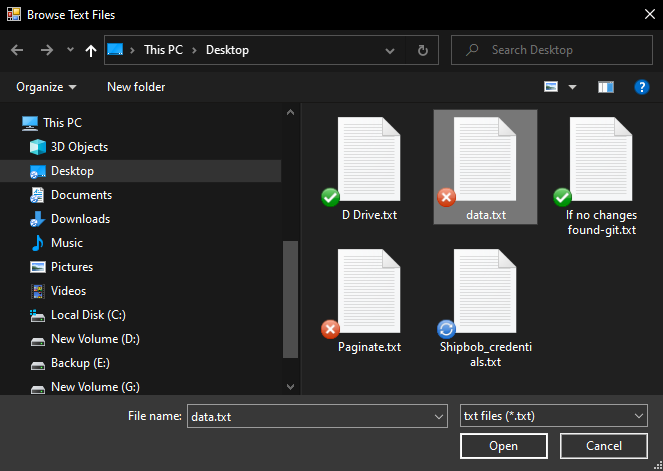


### Load commands

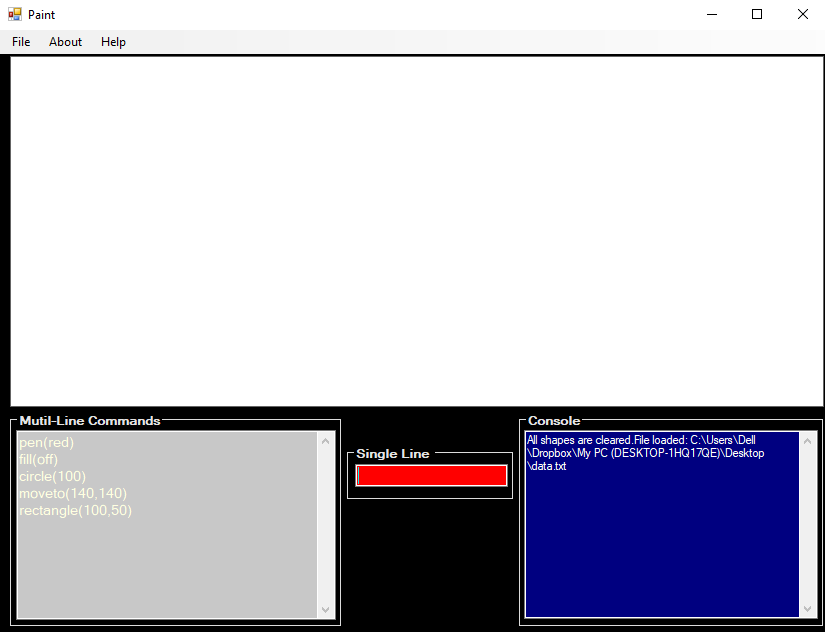
Step 1: Click on load button to open text file containing commands



Step 2: Choose file from open dialog box and click open



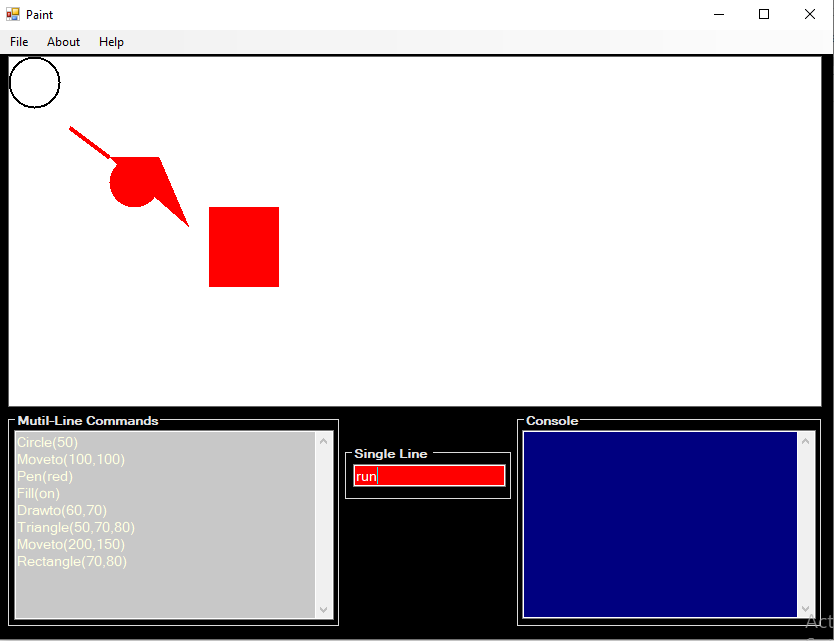
Step 3: File loaded and read successfully



# Command validation check

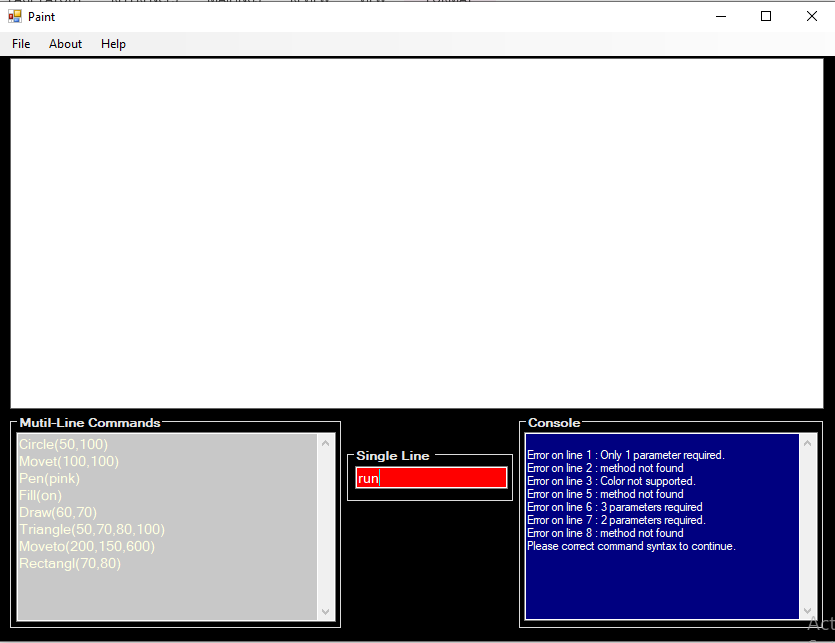
### Valid commands and execution

When valid commands are entered the commands are executed properly with no errors in console.



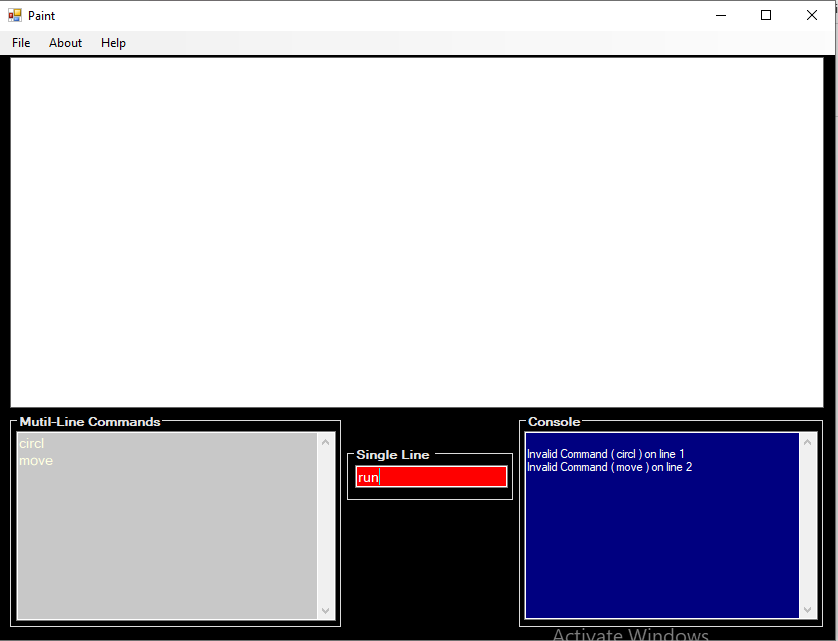
### Invalid Parameter Execution

When invalid commands are entered, error message with respective nature of error is shown in console.



## Invalid Command Execution

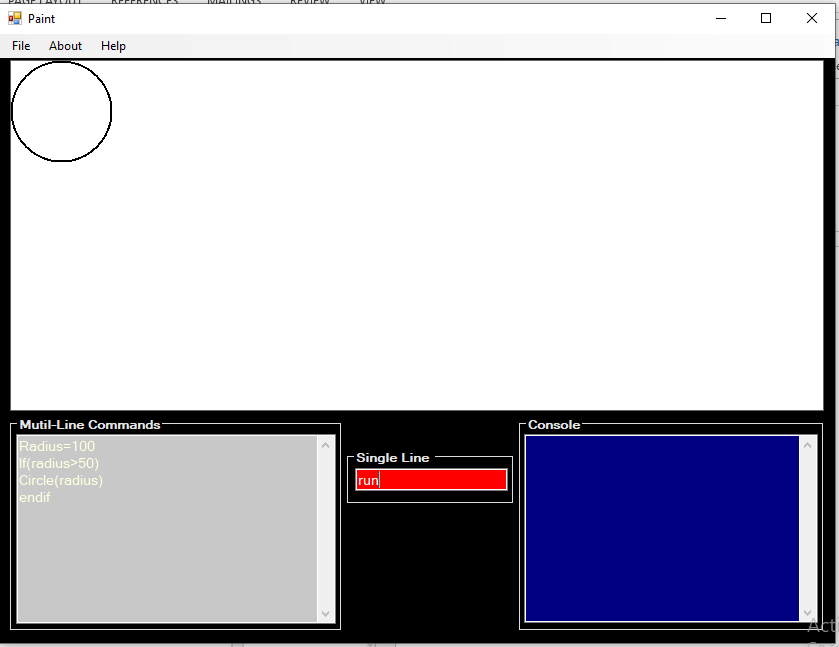
When invalid command is passed then error is shown in console



# If Endif/ Then Command

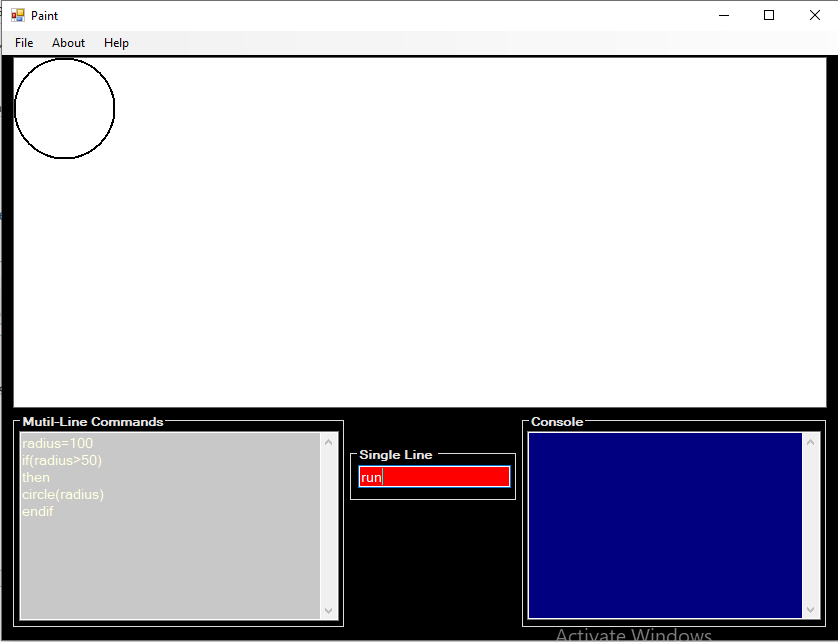
## When the correct code is entered inside if endif

If the condition is matched, the statements inside the if else endif run successfully.

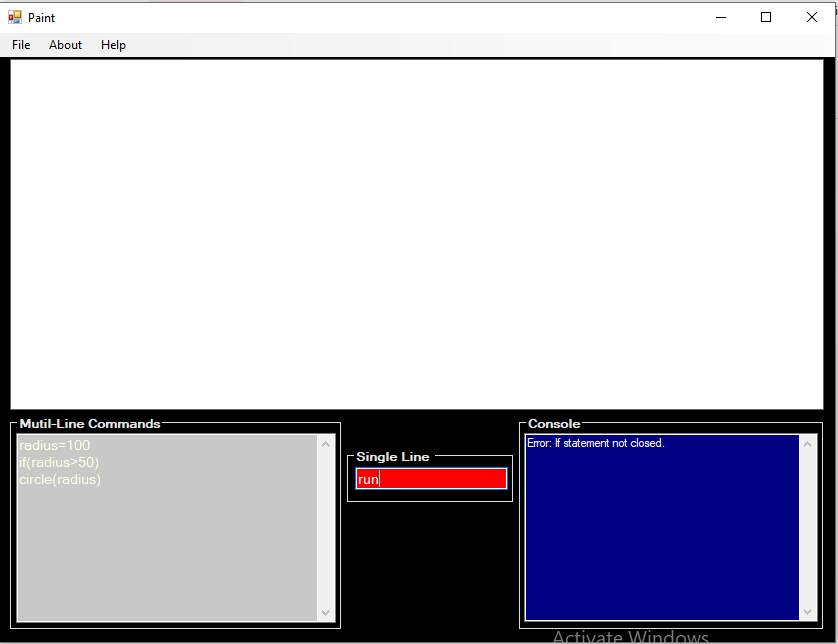


## When the correct code is entered inside if then

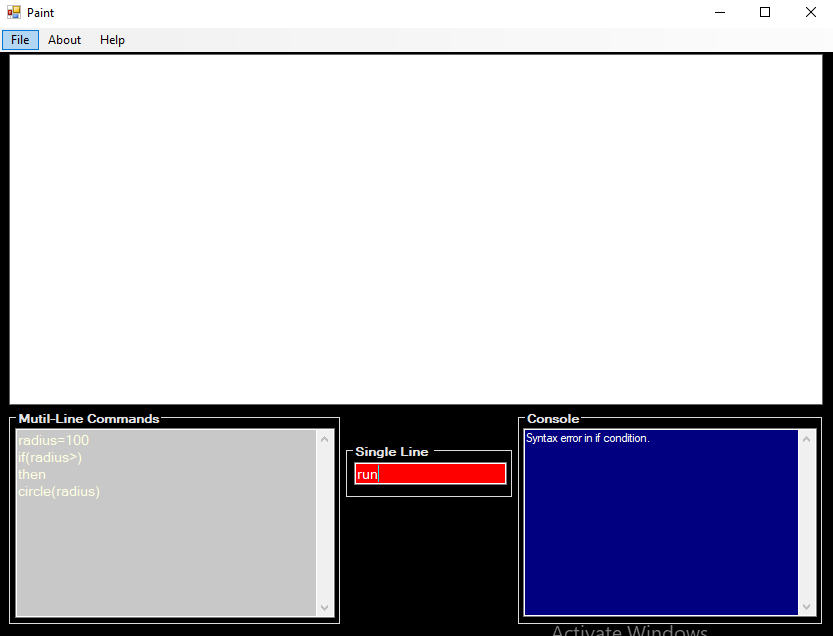
The statement after then is run successfully.



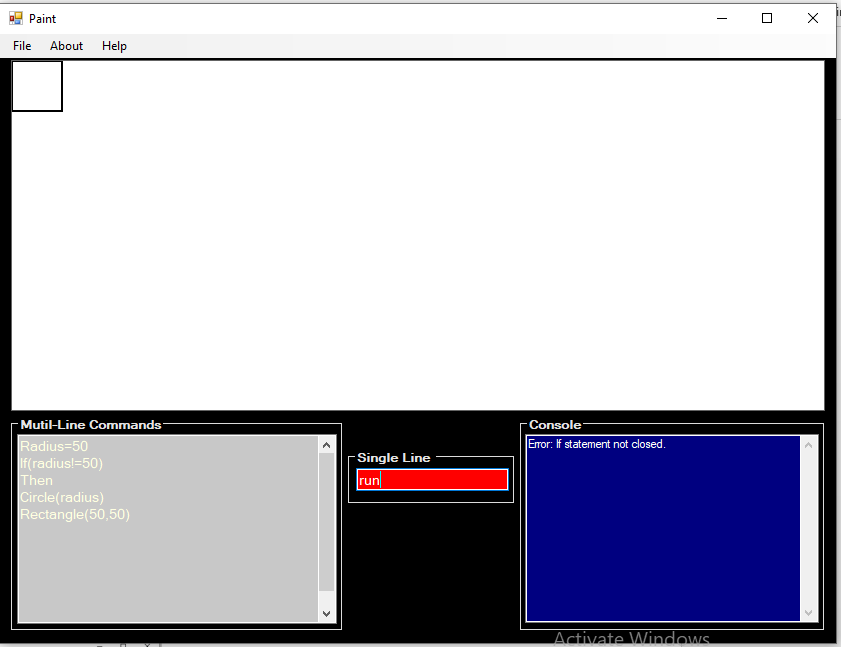
## When the variable is not declared

If the variable is not declared, the error is shown on console.

## When condition inside if is not complete

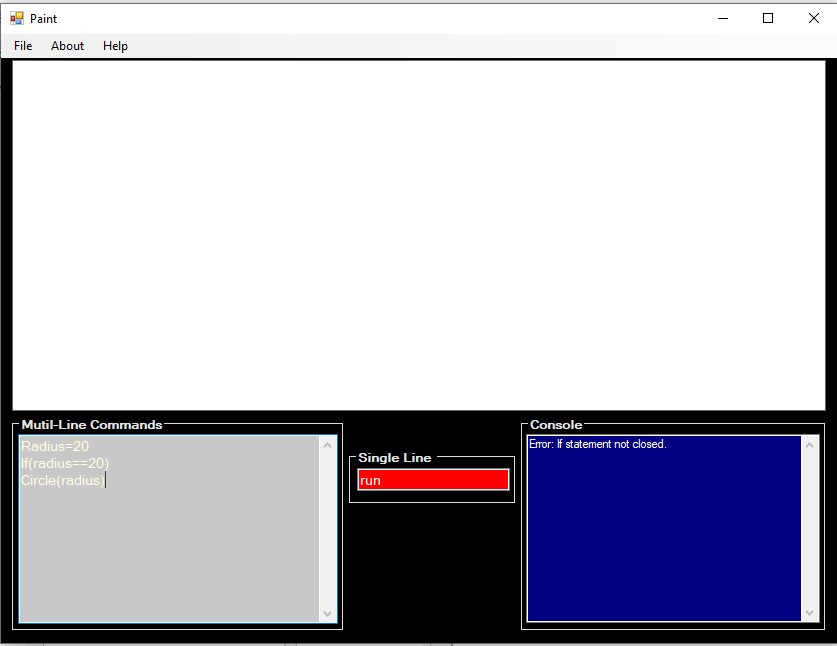


## When condition does not match

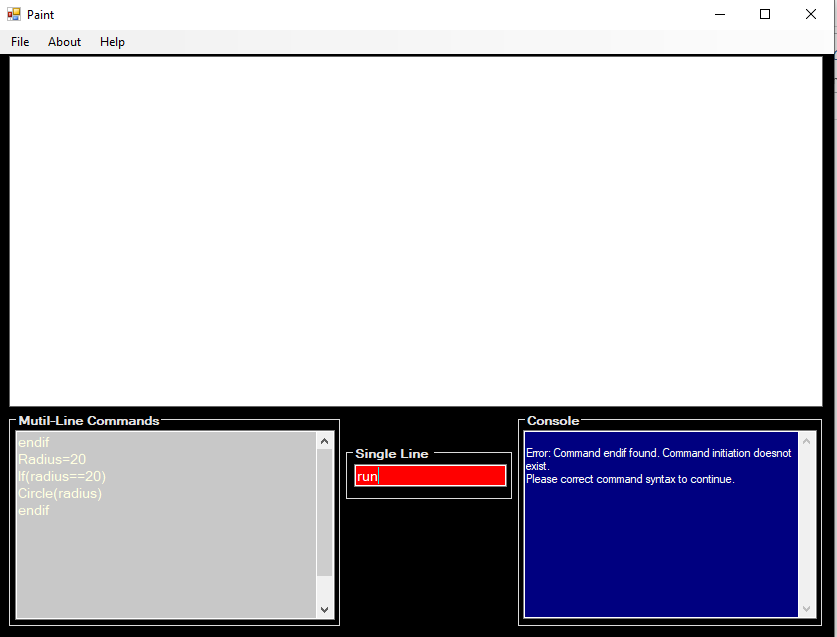


The statement inside the if does not run and other statements are executed.

## When if command is not closed

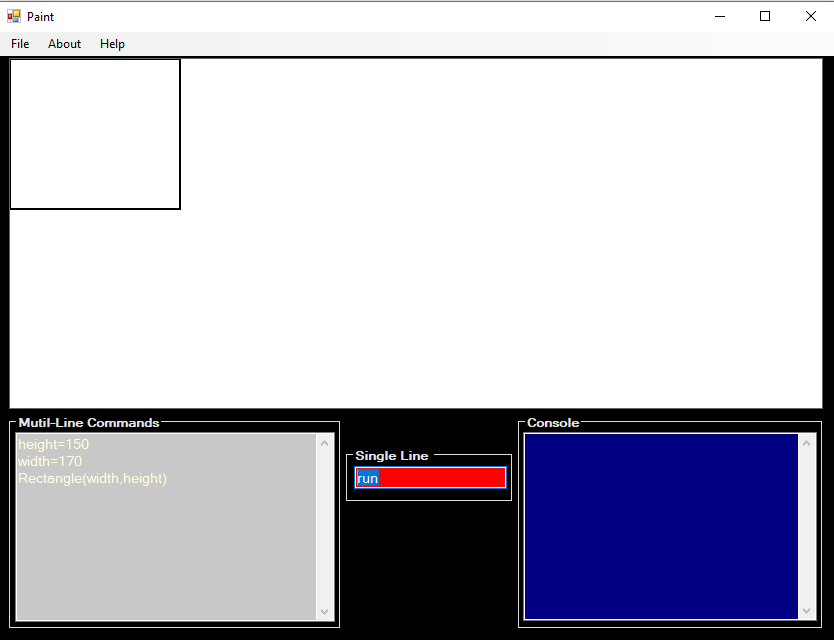


## When endif is found before if command



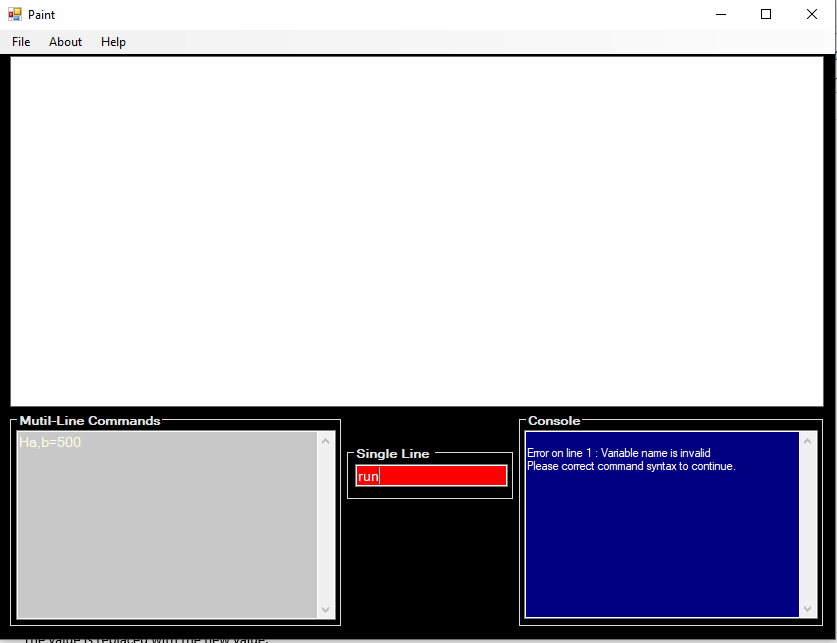
# Variable Command

## When correct variable is declared.

The code runs successfully. 

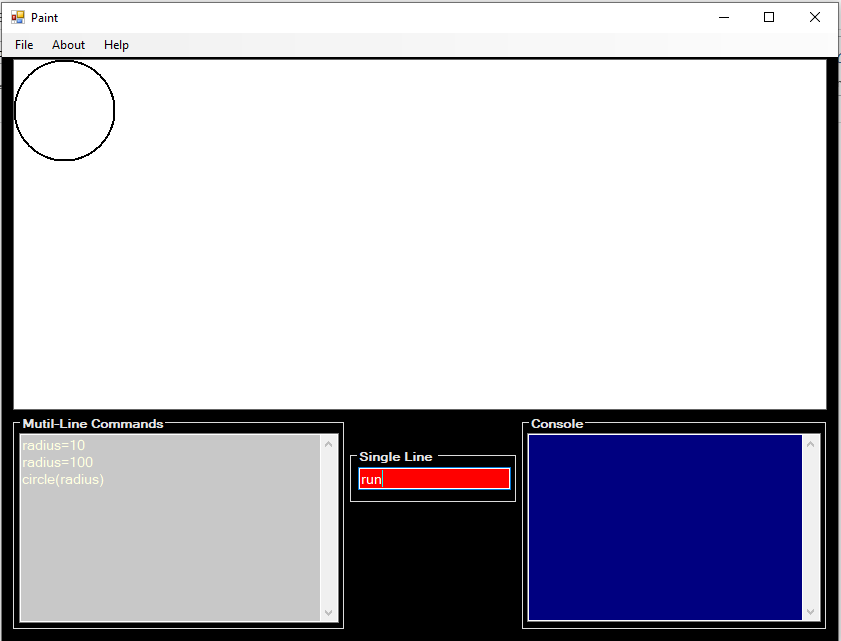
## When the variable is incorrectly declared.

Invalid declaration validation is shown on console.



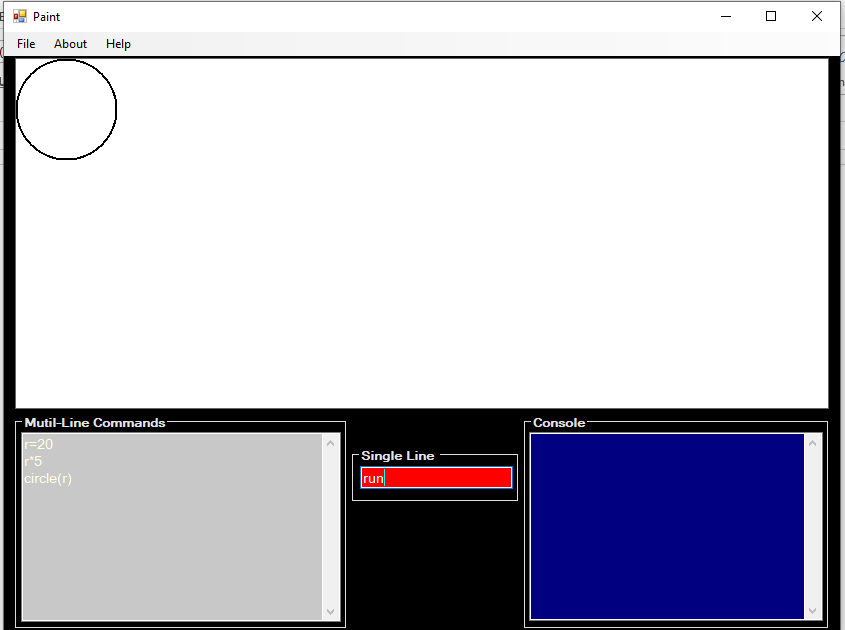
## When the same variable is declared again.

The value is replaced with the new value.



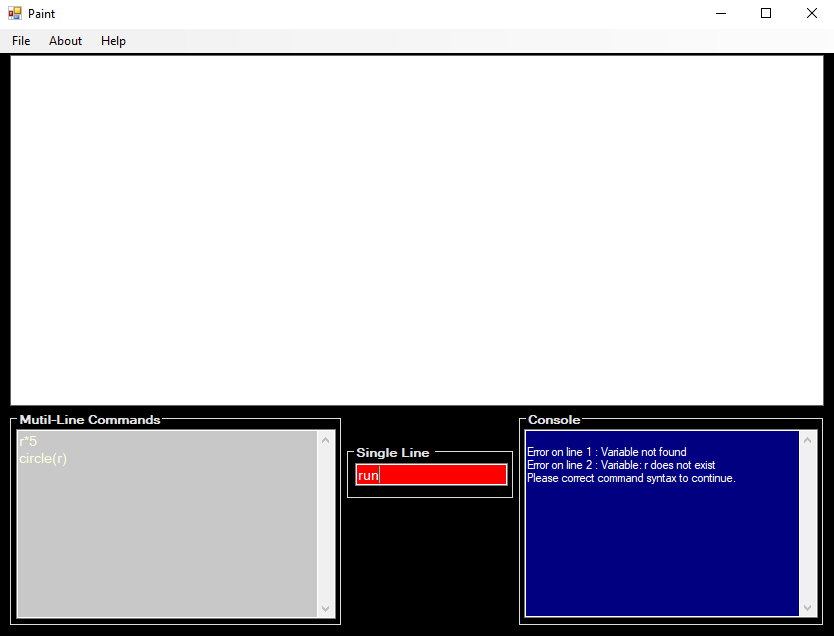
# Variable Operation Command

## When the variable operation is correct.

The value of variable is changed accordingly. 

## When the variable is not present

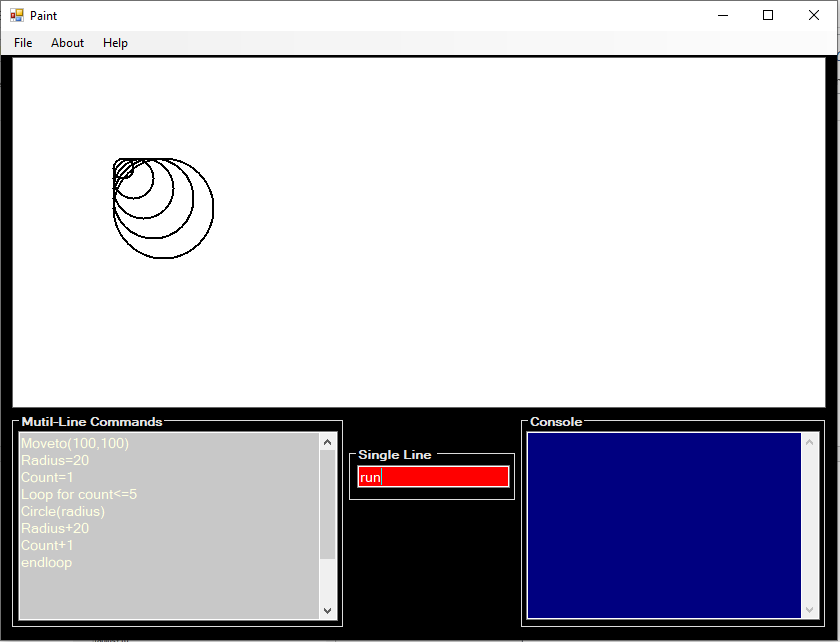
When the variable is not present the validation is shown on the console.



# Loop command

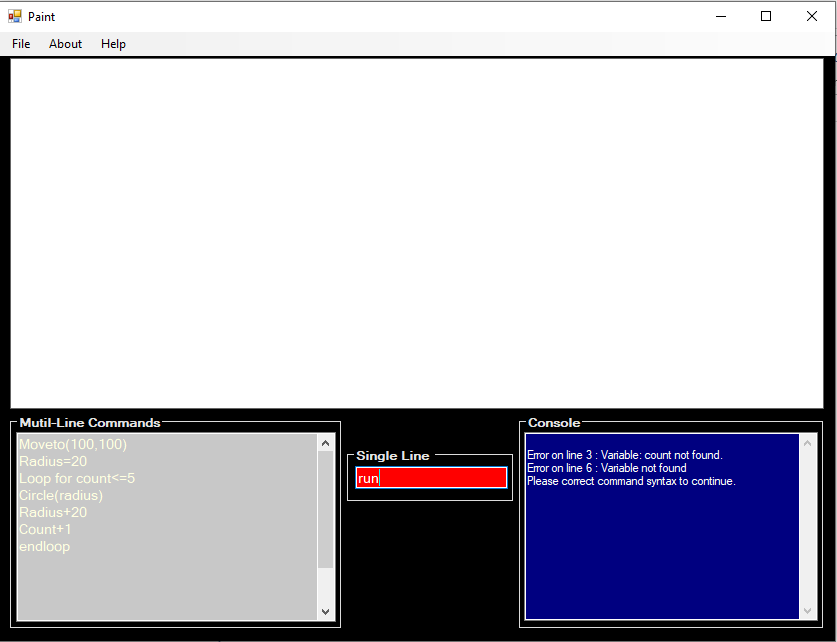
## When valid loop command is run

The statement run successfully.



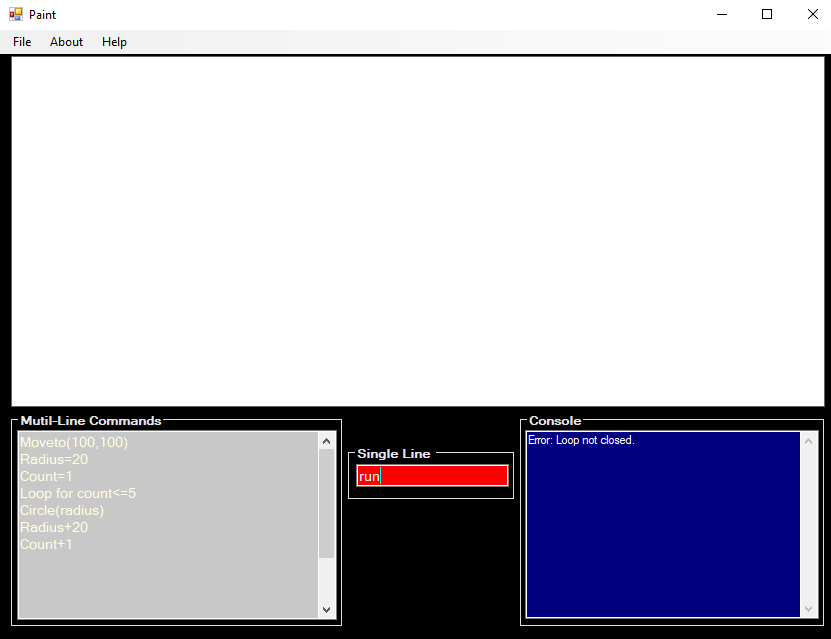
## When the counter variable is not declared.

The error on the declaration of loop is shown in console.



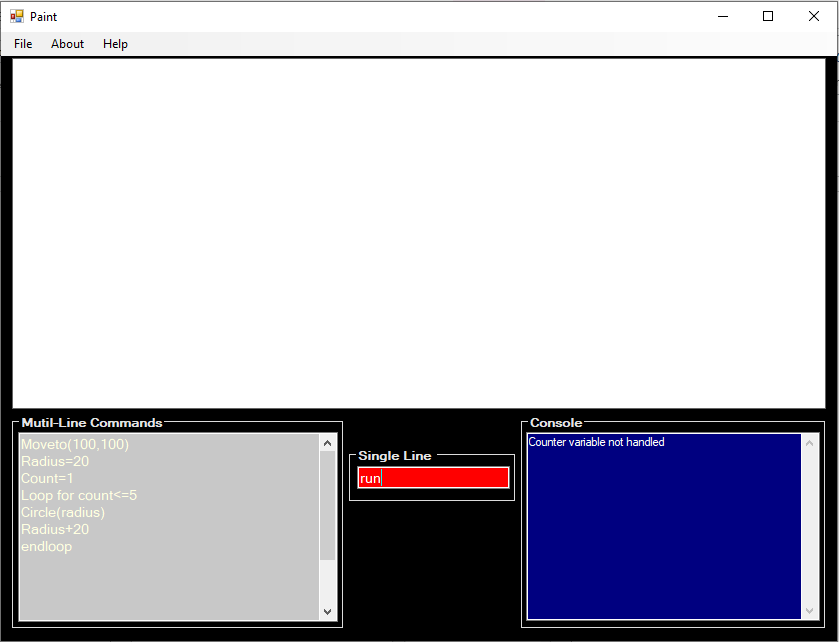
## When the loop is not closed

Validation of loop not closed is shown in the console.



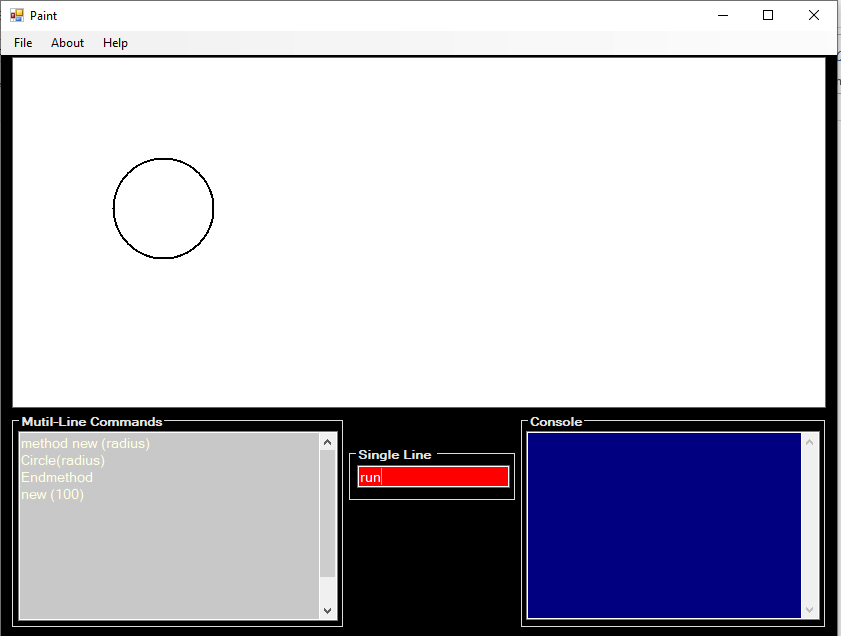
## When the counter variable is not handled

Validation showing that the counter variable is not handled is show in console.



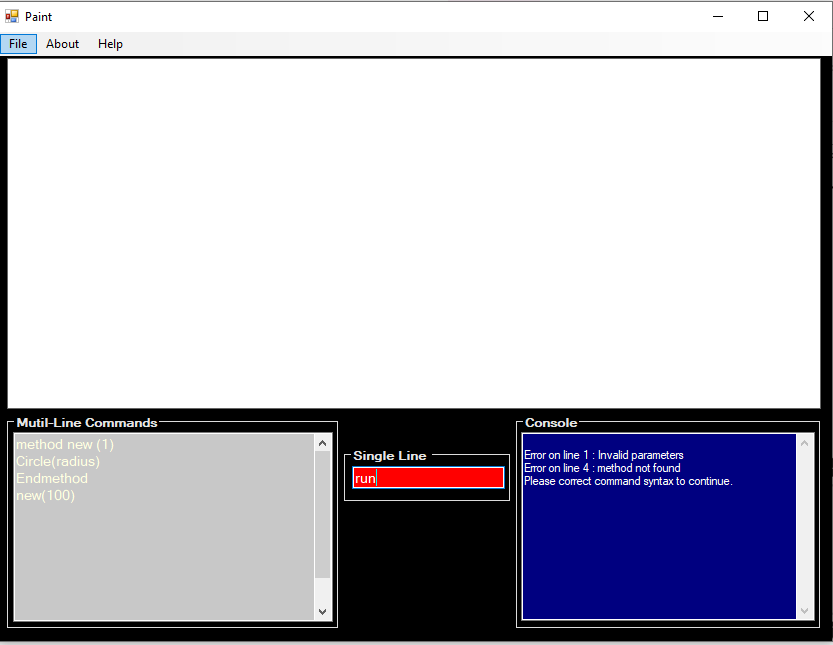
# Method Command

## When the method declaration is valid.

The method called works successfully. 

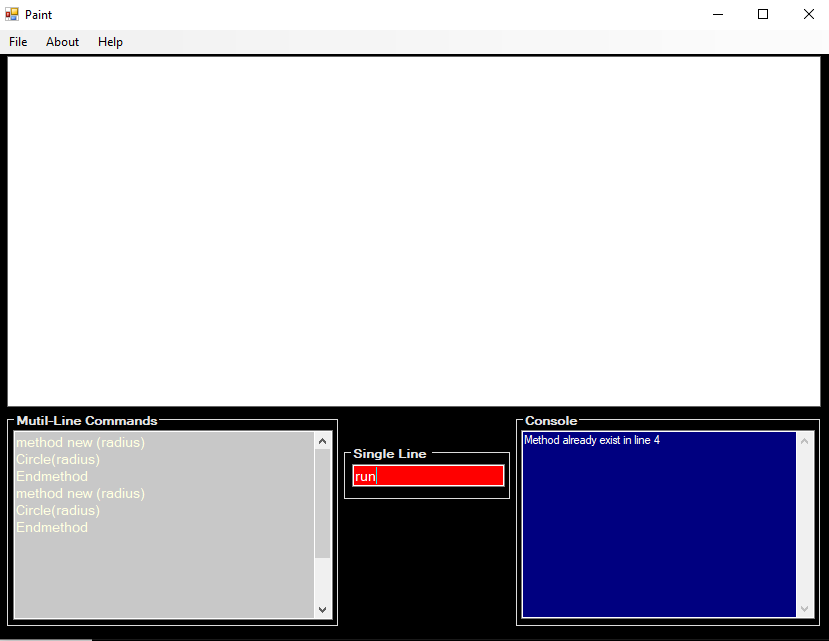
## When parameter is Numeric

Validation showing that the parameter cannot be numeric is shown in console.



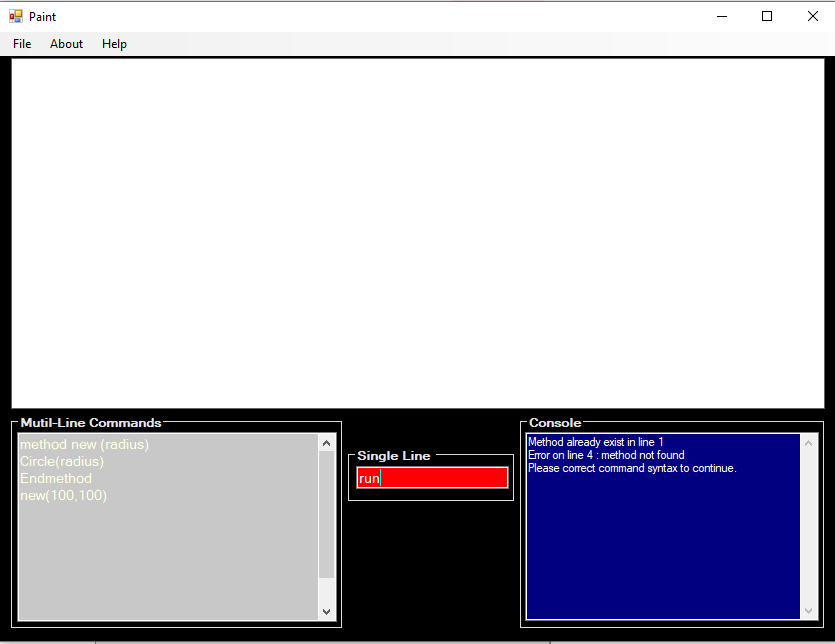
## When same method is declared twice

Validation showing that the method has already been declared and cannot be declared again is displayed in console.



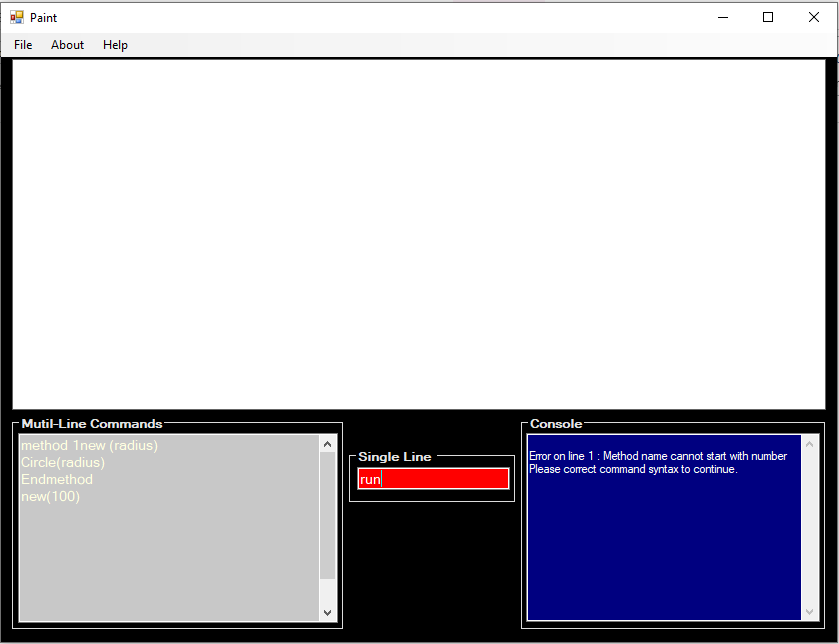
## When method call has different parameters than declaration

Validation showing that the method with similar parameter does not exist is displayed on console.

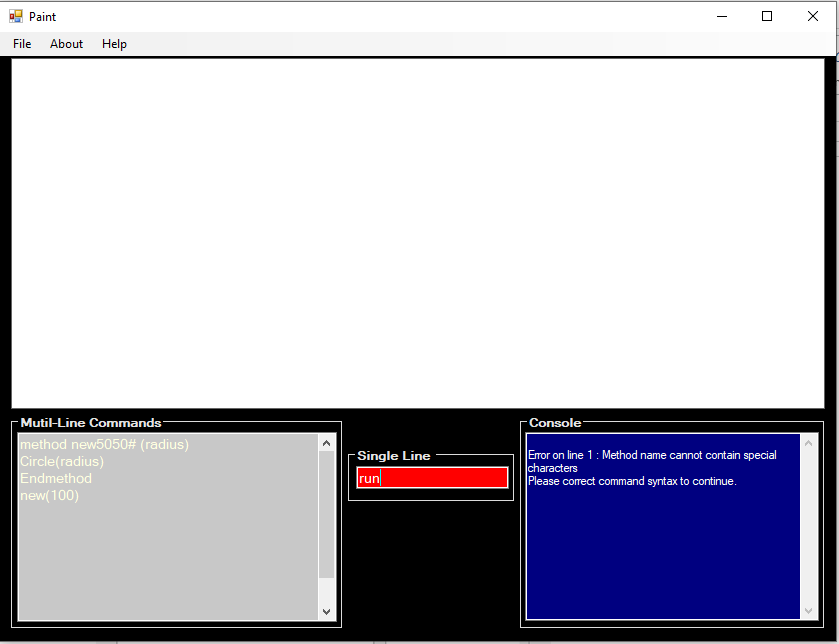


## When the name of method is invalid

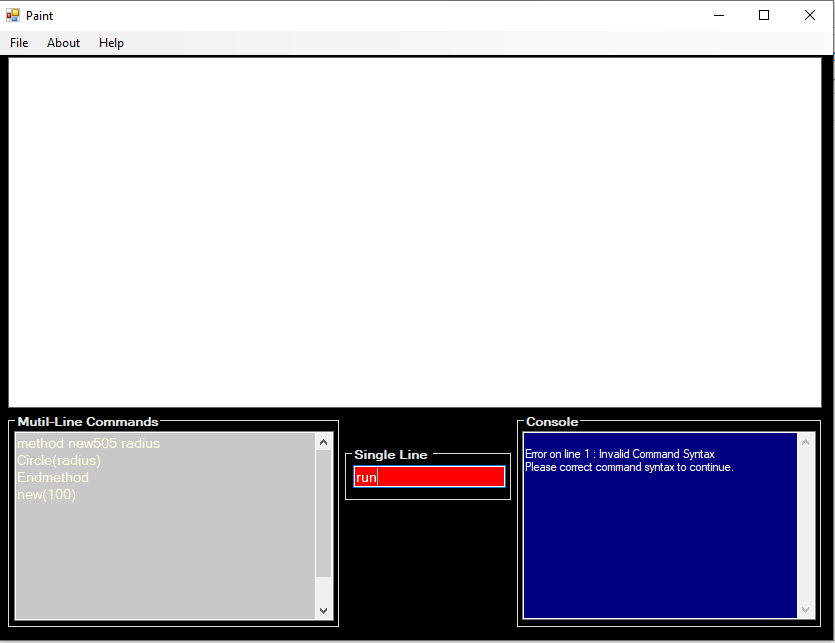
### Method name starting with number



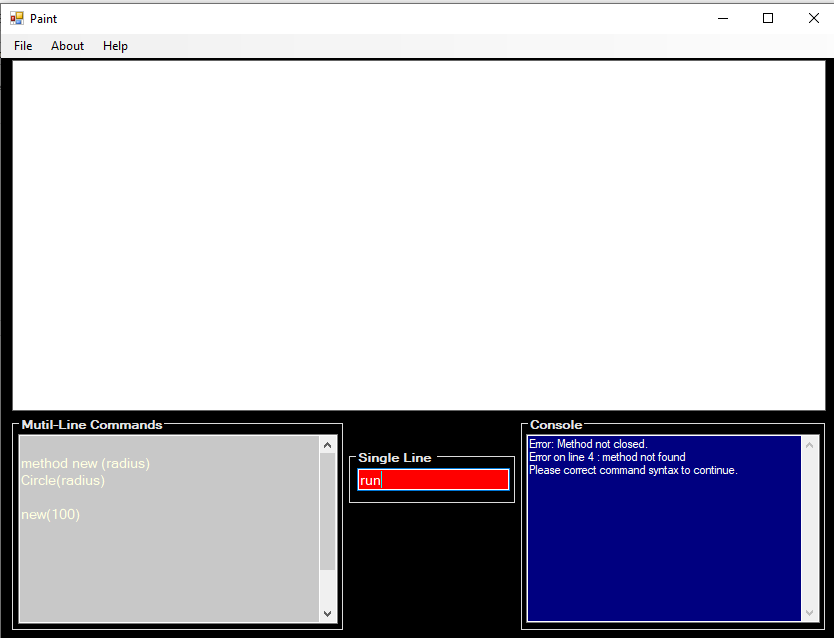
### Method name contains invalid characters



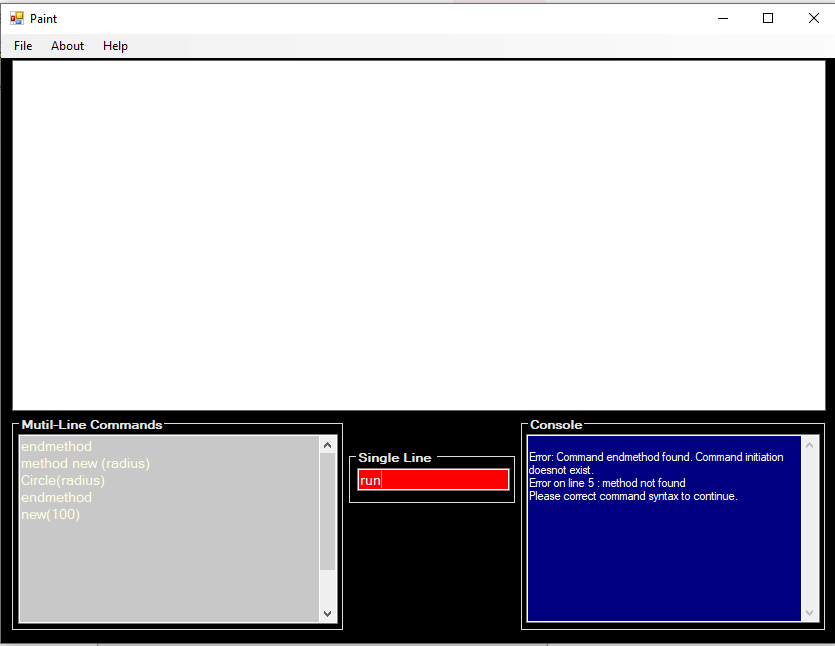
## When method syntax is invalid



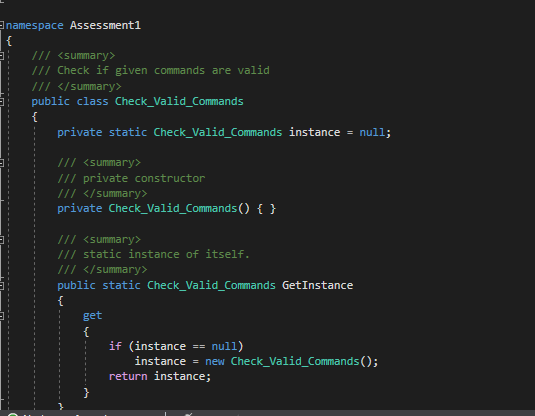
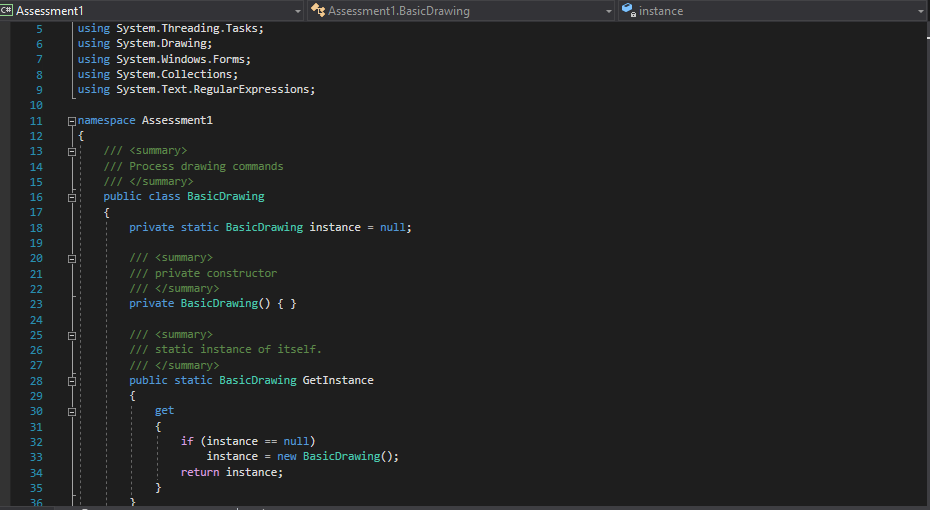
## When method is not closed

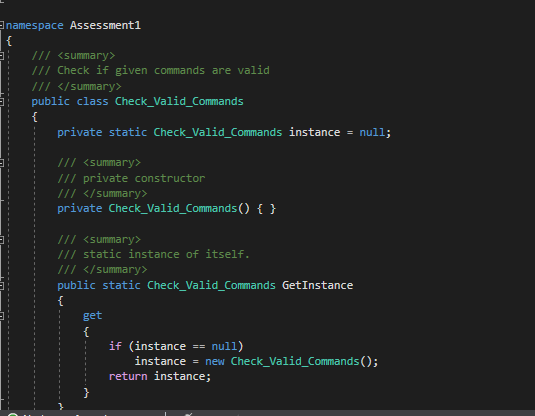


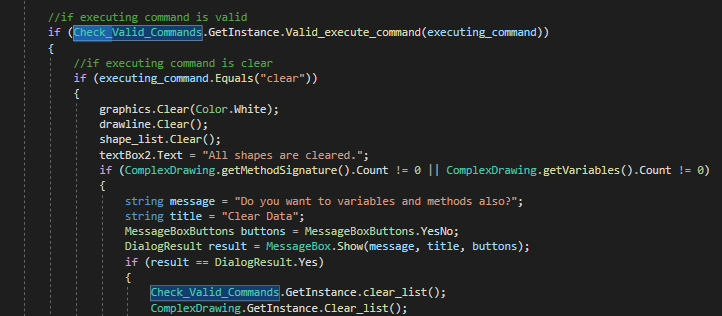
## When method is closed before or without method



# Singleton Design Pattern





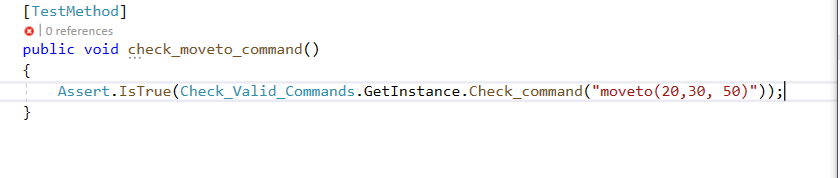
# Unit Testing

## Testing valid drawing commands

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | | **001** | | **Test Case Description** | | | | | | | | **Valid Command Check** | | | | | | | | | | | | | | | | | | | | |
| **Created By** | | | **Rakshya** | | **Review By** | | | | | | | | **Rakshya** | | | | | | | | |  | | |  | | |  | | |
|  |  |  | |  | |  | | | |  | |  | | |  | | |  | |  |  | |  | | |  | | |
|  |  |  | |  | |  | | | |  | |  | | |  | | |  | |  |  | |  | | |  | | |
| **Tester's Name** | | | **Rakshya Lama Moktan** | | | | **Date Tested** | | | | | | | **January 16th 2020** | | | | | **Test Case[Pass/Fail/Not]** | | | | | | **Pass** | | | | | | |
|  |  |  | |  | |  | | | |  | |  | | |  | | |  | |  |  | |  | | |  | | |
|  |  |  | |  | |  | | | |  | |  | | |  | | |  | |  |  | |  | | |  | | |
|  |  |  | |  | |  | | | |  | |  | | |  | | |  | |  |  | |  | | |  | | |
| **S#** | **Prerequisites** | | | | | | | | | |  | | **S#** | | | **Test Data** | | | | | | | |  | | |  | | |
| 1 | Visual Studio | | | | | | | | | |  | | 1 | | | Instance of Check\_Valid\_Commands Class | | | | | | | |  | | |  | | |
| 2 | Unit Testing Class | | | | | | | | | |  | | 2 | | | string[] commands = {“run”, “clear”,”reset”} | | | | | | | |  | | |  | | |
| 3 |  | | | | | | | | | |  | | 3 | | | Moveto (20,30), moveto(20,y), movto(20,30) | | | | | | | |  | | |  | | |
| 4 |  | | | | | | | | | |  | | 4 | | | Drawto (20,30), drawto(20,x), drwtos(20,30) | | | | | | | |  | | |  | | |
|  |  | | | | | | | | | |  | | 5 | | | Fill (on), fill (on,off), filsl (on) | | | | | | | |  | | |  | | |
|  |  | | | | | | | | | |  | | 6 | | | Pen (red), pen(nocolor), pn(red) | | | | | | | |  | | |  | | |
|  |  | | | | | | | | | |  | | 7 | | | Rectangle (20,30), rectangle (2t,30,40), rectanglesc(20,30) | | | | | | | |  | | |  | | |
|  |  | | | | | | | | | |  | | 8 | | | Triangle(20,40,50), triangle(40,50), triang(20,40,50) | | | | | | | |  | | |  | | |
|  |  | | | | | | | | | |  | | 9 | | | Polygon (20,40,50,60), polygon (40,50), polgon(20,30,40,50) | | | | | | | |  | | |  | | |
|  |  | | | | | | | | | |  | | 10 | | | Rotate(20), rotate(xy,200), rotae(20) | | | | | | | |  | | |  | | |
|  |  | | | | | | | | | |  | | 11 | | | Circle (20), circle (20,3y), circ (20) | | | | | | | |  | | |  | | |
|  |  |  | |  | |  | | | |  | |  | | |  | | |  | |  |  | |  | | |  | | |
|  |  |  | |  | |  | | | |  | |  | | |  | | |  | |  |  | |  | | |  | | |
| **Test Scenario** |  |  | | **Commands are passed to command checking methods. Methods validates and returns Boolean value.** | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  |  |  | |  | | | |  | |  | |  | | |  | | |  | |  |  | |  | | |  | | |
|  |  |  | |  | | | |  | |  | |  | | |  | | |  | |  |  | |  | | |  | | |
| **Step #** | **StepDetails** | | | | | | | | **Expected Results** | | | | | | | | **Actual Results** | | | | | **Pass/Fail/Not executed/ Suspended** | | | | | | | | | | |
|
| 1 | Pass variable commands to valid\_execute\_command | | | | | | | | Run = True  Reset = True  Clear = True | | | | | | | | As Expected | | | | | Pass | | | | | | | | | | |
| 2 | Pass moveto commands to check\_command | | | | | | | | Moveto(20,30) = True Moveto(20,y) = False Movto(20,30) = False | | | | | | | | As Expected | | | | | Pass | | | | | | | | | | |
| 3 | Pass drawto commands to check\_command | | | | | | | | Drawto (20,30) = True Drawto(20,x) = False Drwtos(20,30) = False | | | | | | | | As Expected | | | | | Pass | | | | | | | | | | |
| 4 | Pass fill commands to check\_command | | | | | | | | Fill (on) = True Fill (on,off) = False Filsl (on) = False | | | | | | | | As Expected | | | | | Pass | | | | | | | | | | |
| 5 | Pass pen commands to check\_command | | | | | | | | Pen (red) = True  pen(nocolor) = False  pn(red) = False | | | | | | | | As Expected | | | | | Pass | | | | | | | | | | |
| 6 | Pass rectangle command to check\_command | | | | | | | | Rectangle (20,30) = True  rectangle (2t,30,40) =False  rectanglesc(20,30) = False | | | | | | | | As Expected | | | | | Pass | | | | | | | | | | |
| 7 | Pass triangle command to check\_command | | | | | | | | Triangle(20,40,50) = True  triangle(40,50) = False  triang(20,40,50) =False | | | | | | | | As Expected | | | | | Pass | | | | | | | | | | |
| 8 | Pass polygon command to check\_command | | | | | | | | Polygon (20,40,50,60) = True  polygon (40,50) =False  polgon(20,30,40,50) = False | | | | | | | | As Expected | | | | | Pass | | | | | | | | | | |
| 9 | Pass rotate command to check\_command | | | | | | | | Rotate(20) = True  rotate(xy,200) =False  rotae(20) = False | | | | | | | | As Expected | | | | | Pass | | | | | | | | | | |
| 10 | Pass circle command to check\_command | | | | | | | | Circle (20) = True  circle (20,3y) = False  circ (20) = False | | | | | | | | As Expected | | | | | Pass | | | | | | | | | | |

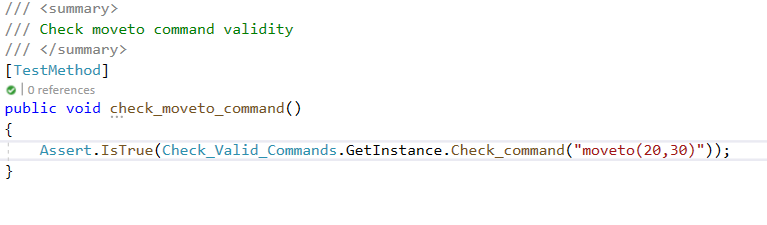
### Working process of this test

#### If Wrong Parameter is Passed, test fails.





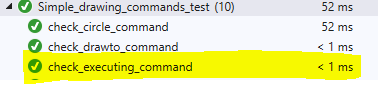
#### If Parameter is Passed Correctly



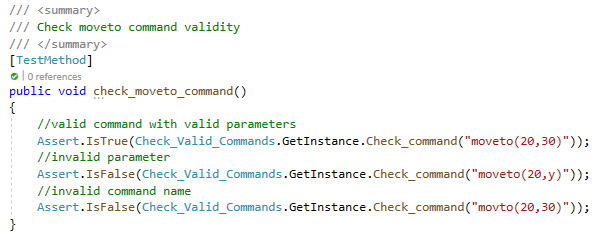
### Screenshots of valid commands test

#### Run, Clear, Reset



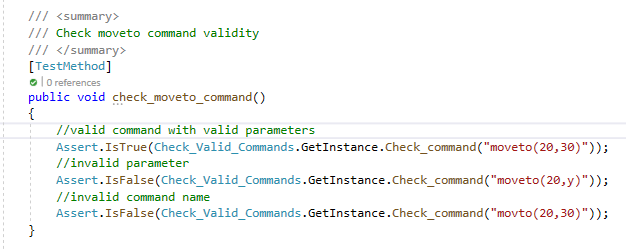


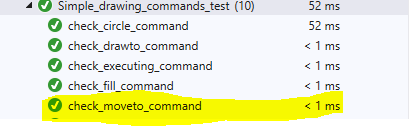
#### MoveTo Command



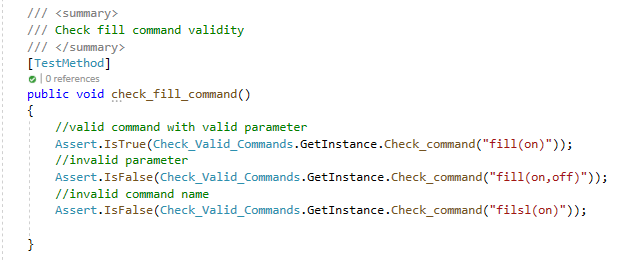


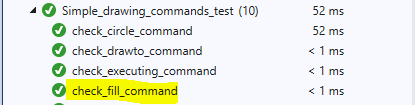
#### DrawTo Command



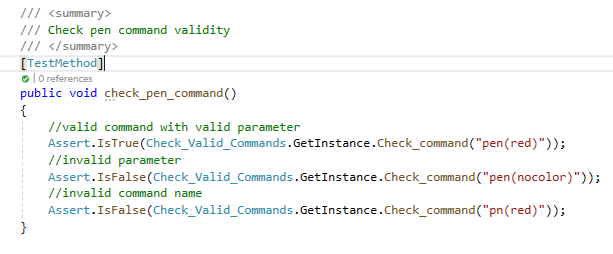


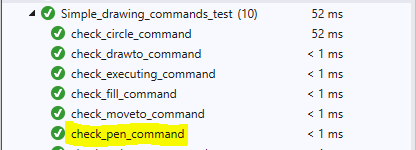
#### Fill



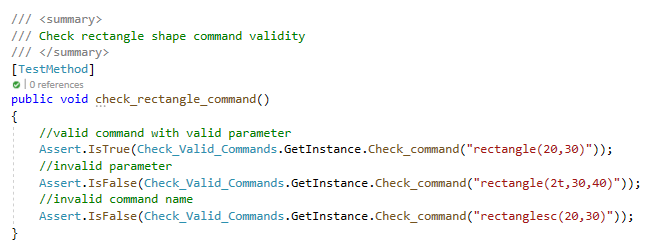


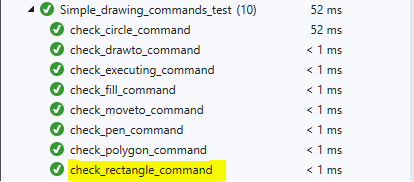
#### Pen



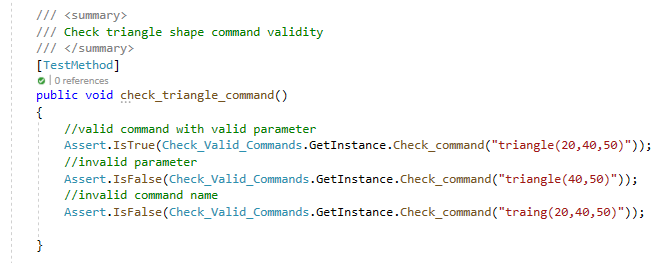


#### Rectangle



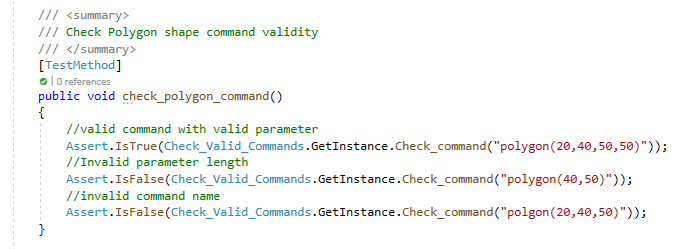


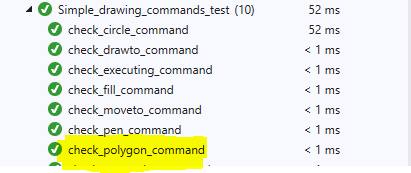
#### Triangle



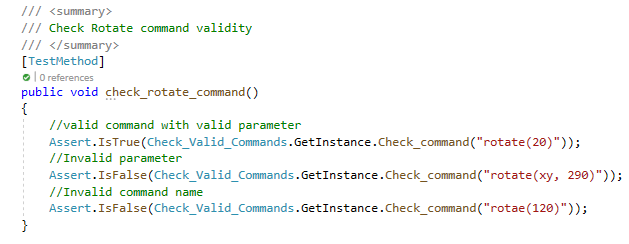


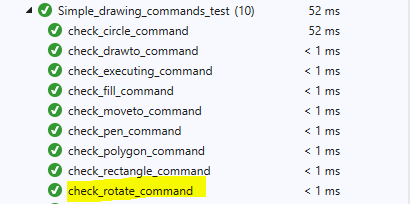
#### Polygon



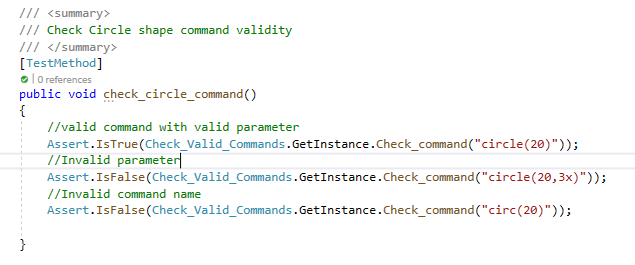


#### Rotate





#### Circle





## If Endif/Then Command

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | | **002** | | **Test Case Description** | | | | | | | | **If Command Check** | | | | | | | | | | | | | | | | | | | | |
| **Created By** | | | **Rakshya** | | **Review By** | | | | | | | | **Rakshya** | | | | | | | | | |  | | |  | | |  | | | | |
|  |  |  | |  | |  | | |  | | |  | | |  | |  | | |  |  |  | | | | |  | | |
|  |  |  | |  | |  | | |  | | |  | | |  | |  | | |  |  |  | | | | |  | | |
| **Tester's Name** | | | **Rakshya Lama Moktan** | | | | **Date Tested** | | | | | | | **January 16th 2020** | | | | **Test Case[Pass/Fail/Not]** | | | | | | | **Pass** | | | | | | |
|  |  |  | |  | |  | | |  | | |  | | |  | |  | | |  |  |  | | | | |  | | |
|  |  |  | |  | |  | | |  | | |  | | |  | |  | | |  |  |  | | | | |  | | |
|  |  |  | |  | |  | | |  | | |  | | |  | |  | | |  |  |  | | | | |  | | |
| **S#** | **Prerequisites** | | | | | | | | |  | | | **S#** | | | **Test Data** | | | | | | | |  | | | |  | | |
| 1 | Visual Studio | | | | | | | | |  | | | 1 | | | Instance of Check\_Valid\_Commands Class | | | | | | | |  | | | |  | | |
| 2 | Unit Testing Class | | | | | | | | |  | | | 2 | | | Instance of ComplexDrawing Class | | | | | | | |  | | | |  | | |
|  |  | | | | | | | | |  | | | 3 | | | Objject of form1 class | | | | | | | |  | | | |  | | |
|  |  | | | | | | | | |  | | | 4 | | | String var = “radius = 10” | | | | | | | |  | | | |  | | |
|  |  | | | | | | | | |  | | | 5 | | | string[] text = {  "if (radius==10)\nthen\ncircle (radius)",  "if (radius==10)\ncircle (radius)\nendif"  }; | | | | | | | |  | | | |  | | |
|  |  | | | | | | | | |  | | | 6 | | | String if\_command = “if (radius==10)” | | | | | | | |  | | | |  | | |
|  |  | | | | | | | | |  | | | 7 | | | Int line\_found\_in = 1 | | | | | | | |  | | | |  | | |
|  |  |  | |  | |  | | |  | | |  | | |  | |  | | |  |  |  | | | | |  | | |
|  |  |  | |  | |  | | |  | | |  | | |  | |  | | |  |  |  | | | | |  | | |
| **Test Scenario** |  |  | | **Commands are passed to command checking methods. Methods validates and returns Boolean value.** | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  |  |  | |  | | | |  |  | | |  | | |  | |  | | |  |  |  | | | | |  | | |
|  |  |  | |  | | | |  |  | | |  | | |  | |  | | |  |  |  | | | | |  | | |
| **Step #** | **StepDetails** | | | | | | | | | | **Expected Results** | | | | | | | | **Actual Results** | | | | **Pass/Fail/Not executed/ Suspended** | | | | | | | | | |
|
| 1 | Pass variable var to Check\_Valid\_Commands.GetInstace.check\_variable | | | | | | | | | | Method returns true | | | | | | | | As Expected | | | | Pass | | | | | | | | | |
| 2 | Pass variable var to ComplexDrawing.GetInstace.run\_variable\_command | | | | | | | | | | Variable is created and method returns true | | | | | | | | As Expected | | | | Pass | | | | | | | | | |
| 3 | Pass each element in array to Check\_Valid\_Commands.Check\_if\_command | | | | | | | | | | Method returns true | | | | | | | | As Expected | | | | Pass | | | | | | | | | |
| 4 | Pass each element in array, array, line\_found\_in, form1 object to ComplexDrawing.GetInstance.run\_if\_command | | | | | | | | | | Method returns true | | | | | | | | As Expected | | | | Pass | | | | | | | | | |

### Testing Code



### Test Summary



Test is successful.

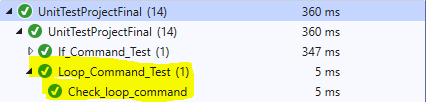
## Loop Command

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | | **003** | | **Test Case Description** | | | | | | | | **Loop Command Check** | | | | | | | | | | | | | | | | | | | | |
| **Created By** | | | **Rakshya** | | **Review By** | | | | | | | | **Rakshya** | | | | | | | | | |  | | |  | | |  | | | | |
|  |  |  | |  | |  | | |  | | |  | | |  | |  | | |  |  |  | | | | |  | | |
|  |  |  | |  | |  | | |  | | |  | | |  | |  | | |  |  |  | | | | |  | | |
| **Tester's Name** | | | **Rakshya Lama Moktan** | | | | **Date Tested** | | | | | | | **January 16th 2020** | | | | **Test Case[Pass/Fail/Not]** | | | | | | | **Pass** | | | | | | |
|  |  |  | |  | |  | | |  | | |  | | |  | |  | | |  |  |  | | | | |  | | |
|  |  |  | |  | |  | | |  | | |  | | |  | |  | | |  |  |  | | | | |  | | |
|  |  |  | |  | |  | | |  | | |  | | |  | |  | | |  |  |  | | | | |  | | |
| **S#** | **Prerequisites** | | | | | | | | |  | | | **S#** | | | **Test Data** | | | | | | | |  | | | |  | | |
| 1 | Visual Studio | | | | | | | | |  | | | 1 | | | Instance of Check\_Valid\_Commands Class | | | | | | | |  | | | |  | | |
| 2 | Unit Testing Class | | | | | | | | |  | | | 2 | | | Instance of ComplexDrawing Class | | | | | | | |  | | | |  | | |
|  |  | | | | | | | | |  | | | 3 | | | Object of form1 class | | | | | | | |  | | | |  | | |
|  |  | | | | | | | | |  | | | 4 | | | string[] variable = { "count = 1", "radius = 20" }; | | | | | | | |  | | | |  | | |
|  |  | | | | | | | | |  | | | 5 | | | string text = "loop for count <=5\ncircle (radius)\nradius+10\ncount+1\nendloop"; | | | | | | | |  | | | |  | | |
|  |  | | | | | | | | |  | | | 6 | | | string loop\_command = "loop for count <=5"; | | | | | | | |  | | | |  | | |
|  |  | | | | | | | | |  | | | 7 | | | int line\_found\_in = 1; | | | | | | | |  | | | |  | | |
|  |  |  | |  | |  | | |  | | |  | | |  | |  | | |  |  |  | | | | |  | | |
|  |  |  | |  | |  | | |  | | |  | | |  | |  | | |  |  |  | | | | |  | | |
| **Test Scenario** |  |  | | **Commands are passed to command checking methods. Methods validates and returns Boolean value.** | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  |  |  | |  | | | |  |  | | |  | | |  | |  | | |  |  |  | | | | |  | | |
|  |  |  | |  | | | |  |  | | |  | | |  | |  | | |  |  |  | | | | |  | | |
| **Step #** | **StepDetails** | | | | | | | | | | **Expected Results** | | | | | | | | **Actual Results** | | | | **Pass/Fail/Not executed/ Suspended** | | | | | | | | | |
|
| 1 | Pass variable var to Check\_Valid\_Commands.GetInstace.check\_variable | | | | | | | | | | Method returns true | | | | | | | | As Expected | | | | Pass | | | | | | | | | |
| 2 | Pass variable var to ComplexDrawing.GetInstace.run\_variable\_command | | | | | | | | | | Variable is created and method returns true | | | | | | | | As Expected | | | | Pass | | | | | | | | | |
| 3 | Pass text to Check\_Valid\_Commands.check\_loop | | | | | | | | | | Method returns true | | | | | | | | As Expected | | | | Pass | | | | | | | | | |
| 4 | Pass variable loop\_command, text, line\_found\_in, form1 object to ComplexDrawing.GetInstance.run\_loop\_command | | | | | | | | | | Method returns true | | | | | | | | As Expected | | | | Pass | | | | | | | | | |

### Testing Code



### Test Summary

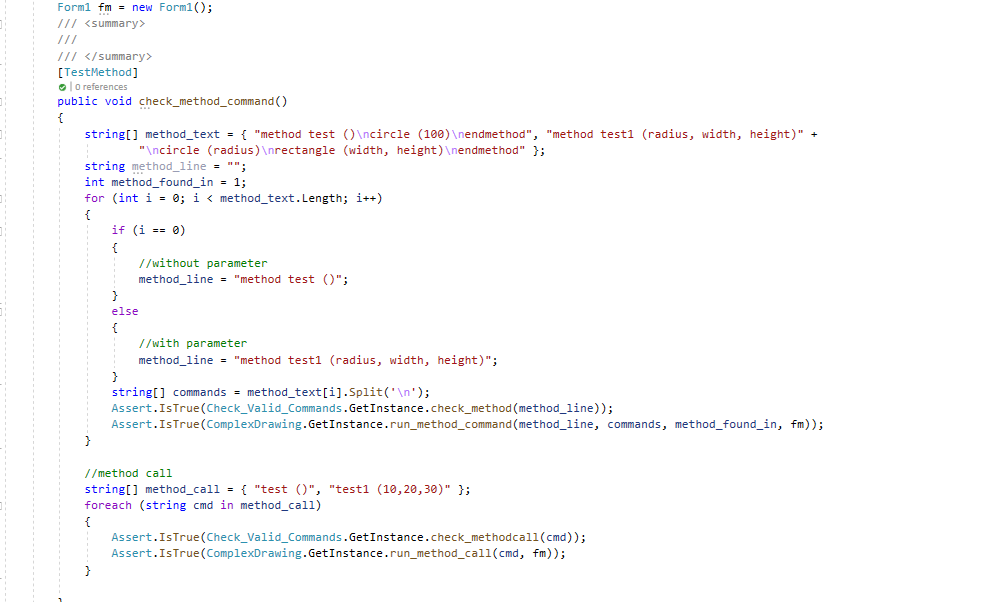


Test is successful.

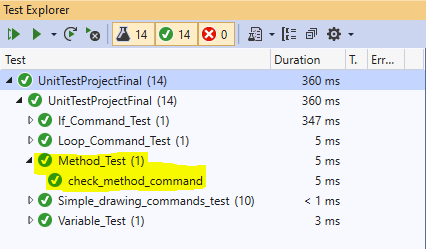
## Method Command

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | | **004** | | **Test Case Description** | | | | | | | **Method Command Check** | | | | | | | | | | | | | | | | | | | | | |
| **Created By** | | | **Rakshya** | | **Review By** | | | | | | | **Rakshya** | | | | | | | | | | |  | | |  | | |  | | | | |
|  |  |  | |  | |  | | |  | |  | | |  | | |  | | |  |  |  | | | | |  | | |
|  |  |  | |  | |  | | |  | |  | | |  | | |  | | |  |  |  | | | | |  | | |
| **Tester's Name** | | | **Rakshya Lama Moktan** | | | | **Date Tested** | | | | | | **January 16th 2020** | | | | | **Test Case[Pass/Fail/Not]** | | | | | | | **Pass** | | | | | | |
|  |  |  | |  | |  | | |  | |  | | |  | | |  | | |  |  |  | | | | |  | | |
|  |  |  | |  | |  | | |  | |  | | |  | | |  | | |  |  |  | | | | |  | | |
|  |  |  | |  | |  | | |  | |  | | |  | | |  | | |  |  |  | | | | |  | | |
| **S#** | **Prerequisites** | | | | | | | | |  | | **S#** | | | **Test Data** | | | | | | | | |  | | | |  | | |
| 1 | Visual Studio | | | | | | | | |  | | 1 | | | Instance of Check\_Valid\_Commands Class | | | | | | | | |  | | | |  | | |
| 2 | Unit Testing Class | | | | | | | | |  | | 2 | | | Instance of ComplexDrawing Class | | | | | | | | |  | | | |  | | |
|  |  | | | | | | | | |  | | 3 | | | Object of form1 class | | | | | | | | |  | | | |  | | |
|  |  | | | | | | | | |  | | 4 | | | string[] method\_text = { "method test ()\ncircle (100)\nendmethod", "method test1 (radius, width, height)\ncircle (radius)\nrectangle (width, height)\nendmethod" }; | | | | | | | | |  | | | |  | | |
|  |  | | | | | | | | |  | | 5 | | | String method\_line = "method test ()"; | | | | | | | | |  | | | |  | | |
|  |  | | | | | | | | |  | | 6 | | | String method\_line = "method test1 (radius, width, height)"; | | | | | | | | |  | | | |  | | |
|  |  | | | | | | | | |  | | 7 | | | int method\_found\_in = 1; | | | | | | | | |  | | | |  | | |
|  |  | | | | | | | | |  | | 8 | | | string[] method\_call = { "test ()", "test1 (10,20,30)" }; | | | | | | | | |  | | | |  | | |
|  |  |  | |  | |  | | |  | |  | | |  | | |  | | |  |  |  | | | | |  | | |
|  |  |  | |  | |  | | |  | |  | | |  | | |  | | |  |  |  | | | | |  | | |
| **Test Scenario** |  |  | | **Commands are passed to command checking methods. Methods validates and returns Boolean value.** | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  |  |  | |  | | | |  |  | |  | | |  | | |  | | |  |  |  | | | | |  | | |
|  |  |  | |  | | | |  |  | |  | | |  | | |  | | |  |  |  | | | | |  | | |
| **Step #** | **StepDetails** | | | | | | | | | | | | | | | **Expected Results** | | | **Actual Results** | | | | **Pass/Fail/Not executed/ Suspended** | | | | | | | | | |
|
| 1 | Pass variable method\_line to Check\_Valid\_Commands.GetInstance.check\_method | | | | | | | | | | | | | | | Method returns true | | | As Expected | | | | Pass | | | | | | | | | |
| 2 | Pass variable method\_line, command list, method\_found\_in, form1 object to ComplexDrawing.GetInstance.run\_method\_command | | | | | | | | | | | | | | | Method returns true | | | As Expected | | | | Pass | | | | | | | | | |
| 3 | Pass method\_call to Check\_Valid\_Commands.GetInstance.check\_methodcall | | | | | | | | | | | | | | | Method returns true | | | As Expected | | | | Pass | | | | | | | | | |
| 4 | Pass method\_call, form1 object to ComplexDrawing.GetInstance.run\_method\_call | | | | | | | | | | | | | | | Method returns true | | | As Expected | | | | Pass | | | | | | | | | |

### Test Code



### Test Summary



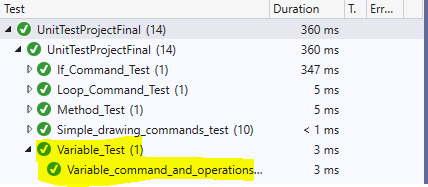
## Variable Command

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | | **005** | | **Test Case Description** | | | | | | | **Variable Command Check** | | | | | | | | | | | | | | | | | | | | | |
| **Created By** | | | **Rakshya** | | **Review By** | | | | | | | **Rakshya** | | | | | | | | | | |  | | |  | | |  | | | | |
|  |  |  | |  | |  | | |  | |  | | |  | | |  | | |  |  |  | | | | |  | | |
|  |  |  | |  | |  | | |  | |  | | |  | | |  | | |  |  |  | | | | |  | | |
| **Tester's Name** | | | **Rakshya Lama Moktan** | | | | **Date Tested** | | | | | | **January 16th 2020** | | | | | **Test Case[Pass/Fail/Not]** | | | | | | | **Pass** | | | | | | |
|  |  |  | |  | |  | | |  | |  | | |  | | |  | | |  |  |  | | | | |  | | |
|  |  |  | |  | |  | | |  | |  | | |  | | |  | | |  |  |  | | | | |  | | |
|  |  |  | |  | |  | | |  | |  | | |  | | |  | | |  |  |  | | | | |  | | |
| **S#** | **Prerequisites** | | | | | | | | |  | | **S#** | | | **Test Data** | | | | | | | | |  | | | |  | | |
| 1 | Visual Studio | | | | | | | | |  | | 1 | | | Instance of Check\_Valid\_Commands Class | | | | | | | | |  | | | |  | | |
| 2 | Unit Testing Class | | | | | | | | |  | | 2 | | | Instance of ComplexDrawing Class | | | | | | | | |  | | | |  | | |
|  |  | | | | | | | | |  | | 3 | | | Object of form1 class | | | | | | | | |  | | | |  | | |
|  |  | | | | | | | | |  | | 4 | | | string[] text = { "width = 10", "rectangle=50", "height= 5" }; | | | | | | | | |  | | | |  | | |
|  |  | | | | | | | | |  | | 5 | | | string[] operation\_text = { "width+10", "rectangle-20", "height\*5", "height/5" }; | | | | | | | | |  | | | |  | | |
|  |  |  | |  | |  | | |  | |  | | |  | | |  | | |  |  |  | | | | |  | | |
|  |  |  | |  | |  | | |  | |  | | |  | | |  | | |  |  |  | | | | |  | | |
| **Test Scenario** |  |  | | **Commands are passed to command checking methods. Methods validates and returns Boolean value.** | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  |  |  | |  | | | |  |  | |  | | |  | | |  | | |  |  |  | | | | |  | | |
|  |  |  | |  | | | |  |  | |  | | |  | | |  | | |  |  |  | | | | |  | | |
| **Step #** | **StepDetails** | | | | | | | | | | | | | | | **Expected Results** | | | **Actual Results** | | | | **Pass/Fail/Not executed/ Suspended** | | | | | | | | | |
|
| 1 | Pass text from string array text to Check\_valid\_Commands.GetInstance.check\_variable | | | | | | | | | | | | | | | Method returns true | | | As Expected | | | | Pass | | | | | | | | | |
| 2 | Pass text from string array text to ComplexDrawing.GetInstance.run\_variable\_command | | | | | | | | | | | | | | | Variable is created and method returns true | | | As Expected | | | | Pass | | | | | | | | | |
| 3 | Pass text from string array operation\_text to Check\_valid\_Commands.GetInstance.check\_variable | | | | | | | | | | | | | | | Method returns true | | | As Expected | | | | Pass | | | | | | | | | |
| 4 | Pass text from string array operation\_text, form1 object to ComplexDrawing.GetInstance.runVariableOperation | | | | | | | | | | | | | | | Variable value are changed and Method returns true | | | As Expected | | | | Pass | | | | | | | | | |

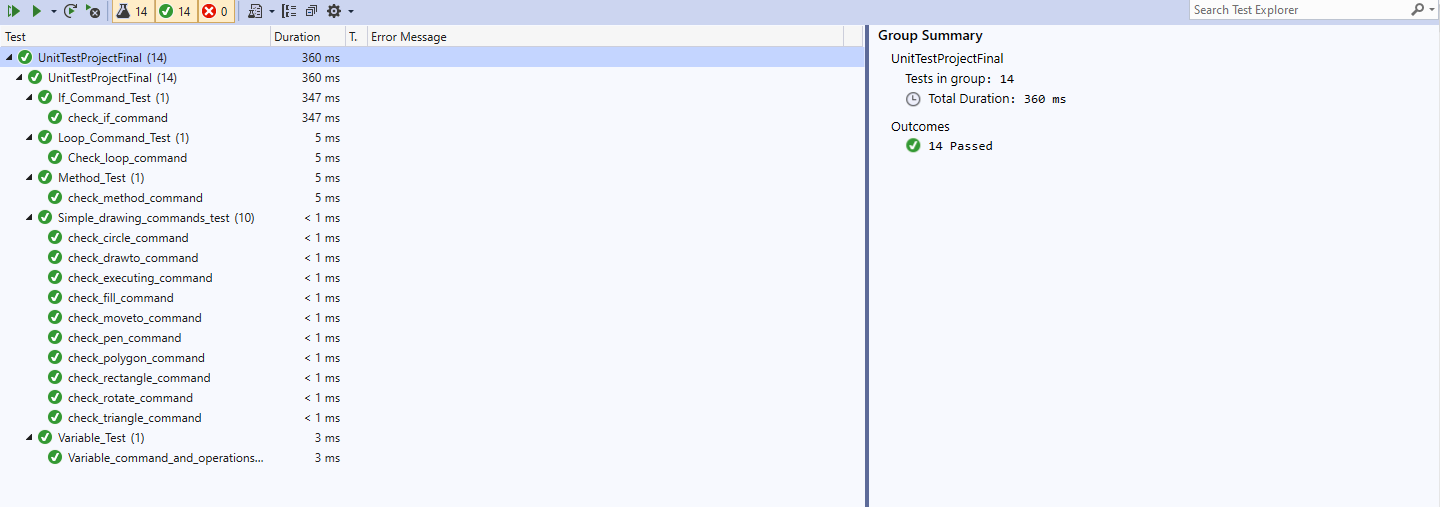
### Test Code



### Test Summary



## Screenshot of all test passed



# Uml Diagram

