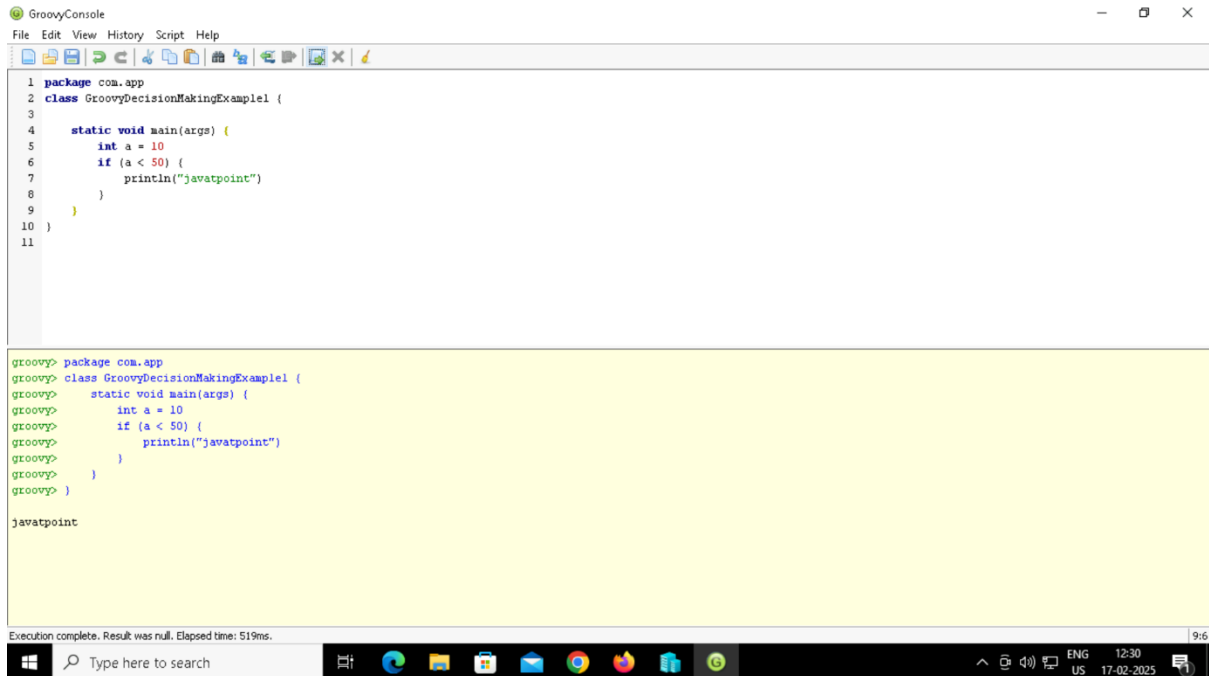


GROOVY PROJECT

1. Groovy program that demonstrates decision-making using an if statement to check if a number is less than 50.



The screenshot shows the GroovyConsole application. The top pane contains the following Groovy code:

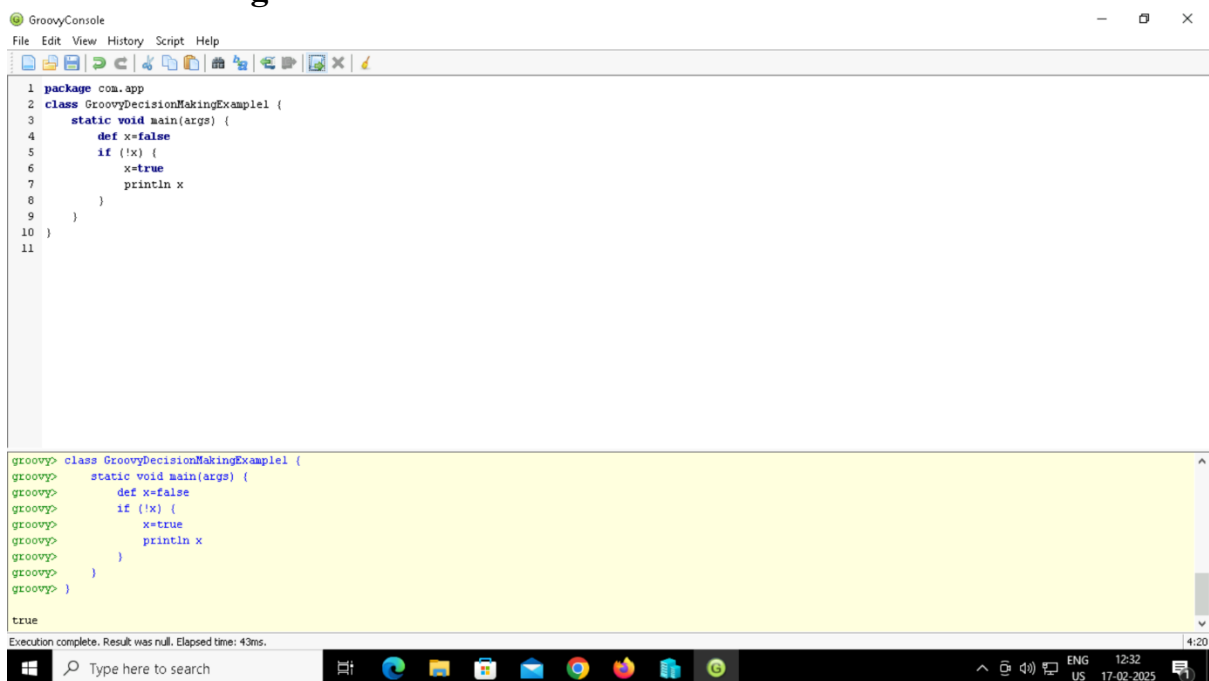
```
1 package com.app
2 class GroovyDecisionMakingExample1 {
3
4     static void main(args) {
5         int a = 10
6         if (a < 50) {
7             println("javatpoint")
8         }
9     }
10 }
11
```

The bottom pane shows the execution output:

```
groovy> package com.app
groovy> class GroovyDecisionMakingExample1 {
groovy>     static void main(args) {
groovy>         int a = 10
groovy>         if (a < 50) {
groovy>             println("javatpoint")
groovy>         }
groovy>     }
groovy> }
javatpoint
```

Execution complete. Result was null. Elapsed time: 519ms.

2. Groovy program that demonstrates the use of logical negation (!) in a decision-making statement.



The screenshot shows the GroovyConsole application. The top pane contains the following Groovy code:

```
1 package com.app
2 class GroovyDecisionMakingExample1 {
3     static void main(args) {
4         def x=false
5         if (!x) {
6             x=true
7             println x
8         }
9     }
10 }
11
```

The bottom pane shows the execution output:

```
groovy> class GroovyDecisionMakingExample1 {
groovy>     static void main(args) {
groovy>         def x=false
groovy>         if (!x) {
groovy>             x=true
groovy>             println x
groovy>         }
groovy>     }
groovy> }
true
```

Execution complete. Result was null. Elapsed time: 43ms.

3. Groovy program that demonstrates the use of an if-else statement for decision-making.

The screenshot shows the GroovyConsole application with a menu bar (File, Edit, View, History, Script, Help) and a toolbar. The script editor contains the following code:

```
1 package com.app
2 class GroovyDecisionMakingExample1 {
3     static void main(args) {
4         int a=20
5         if (a<50) {
6             println ("The value is less than 50");
7         }
8         else {
9             println ("The value is greater than 50");
10        }
11    }
12 }
13
```

The console output shows the code being executed line by line, with the final output being "The value is less than 50". The status bar at the bottom indicates "Execution complete. Result was null. Elapsed time: 20ms." and the system tray shows the date and time as 12:34 on 17-02-2025.

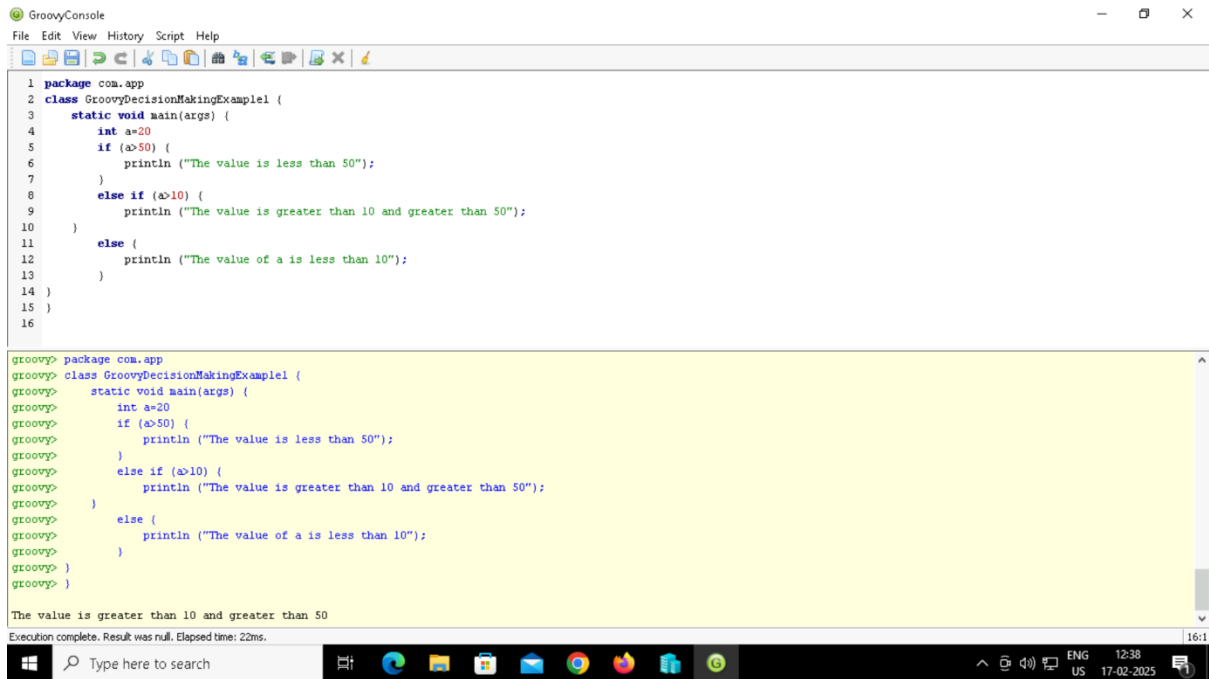
4. Groovy program that demonstrates the use of an if-else statement to check whether a number is positive or negative

The screenshot shows the GroovyConsole application with a menu bar (File, Edit, View, History, Script, Help) and a toolbar. The script editor contains the following code:

```
1 package com.app
2 class GroovyDecisionMakingExample1 {
3     static void main(args) {
4         int a=40
5         if (a>0) {
6             println ("Number is positive");
7         }
8         else {
9             println ("Number is negative");
10        }
11    }
12 }
13
```

The console output shows the code being executed line by line, with the final output being "Number is positive". The status bar at the bottom indicates "Execution complete. Result was null. Elapsed time: 31ms." and the system tray shows the date and time as 12:36 on 17-02-2025.

5. Groovy program that demonstrates the use of an if-else statement



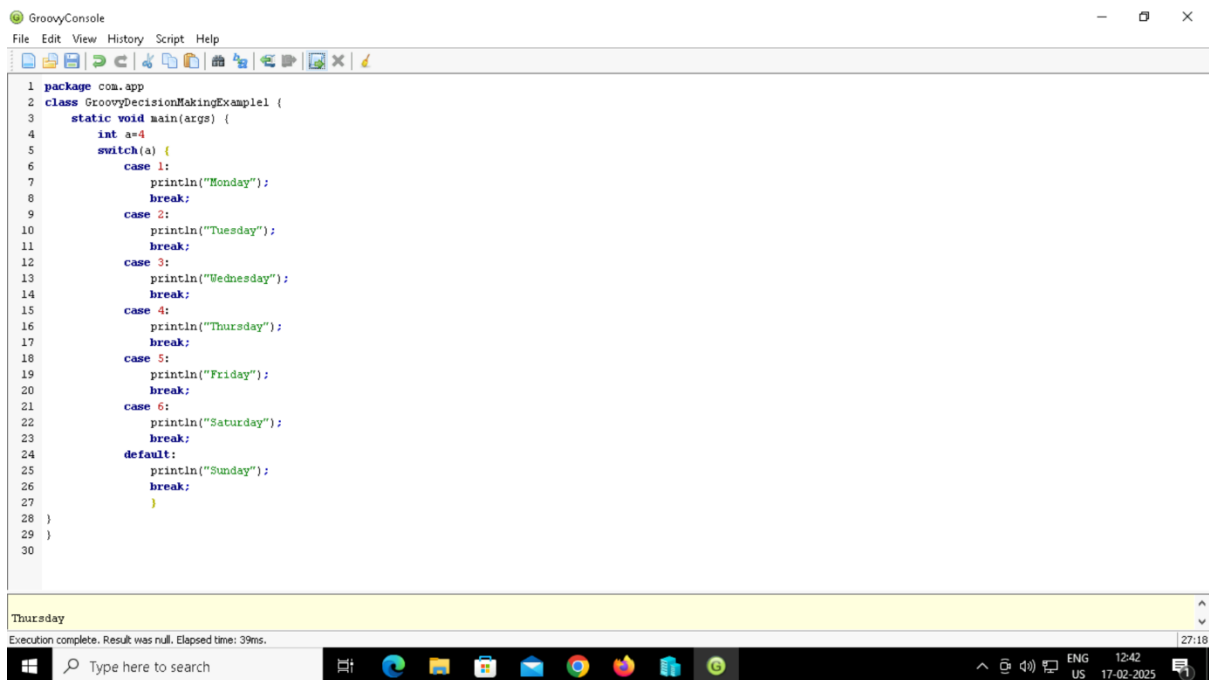
```
1 package com.app
2 class GroovyDecisionMakingExample1 {
3     static void main(args) {
4         int a=20
5         if (a>50) {
6             println ("The value is less than 50");
7         }
8         else if (a>10) {
9             println ("The value is greater than 10 and greater than 50");
10        }
11        else {
12            println ("The value of a is less than 10");
13        }
14    }
15 }
16 }
```

```
groovy> package com.app
groovy> class GroovyDecisionMakingExample1 {
groovy>     static void main(args) {
groovy>         int a=20
groovy>         if (a>50) {
groovy>             println ("The value is less than 50");
groovy>         }
groovy>         else if (a>10) {
groovy>             println ("The value is greater than 10 and greater than 50");
groovy>         }
groovy>         else {
groovy>             println ("The value of a is less than 10");
groovy>         }
groovy>     }
groovy> }
groovy> }
```

The value is greater than 10 and greater than 50

Execution complete. Result was null. Elapsed time: 22ms.

6. Groovy program using Switch Statement



```
1 package com.app
2 class GroovyDecisionMakingExample1 {
3     static void main(args) {
4         int a=4
5         switch(a) {
6             case 1:
7                 println("Monday");
8                 break;
9             case 2:
10                println("Tuesday");
11                break;
12             case 3:
13                println("Wednesday");
14                break;
15             case 4:
16                println("Thursday");
17                break;
18             case 5:
19                println("Friday");
20                break;
21             case 6:
22                println("Saturday");
23                break;
24             default:
25                println("Sunday");
26                break;
27         }
28     }
29 }
30 }
```

```
Thursday
```

Execution complete. Result was null. Elapsed time: 39ms.

7. Groovy program to demonstrate string manipulation by creating and printing a string, along with string concatenation .

The screenshot shows the GroovyConsole application with a menu bar (File, Edit, View, History, Script, Help) and a toolbar. The script editor contains the following code:

```
1 package com.app
2 class GroovyDecisionMakingExample1 {
3     static void main(args) {
4         String s1='Javatpoint'
5         println s1
6         println 'This is tutorial on Groovy at' +s1
7     }
8 }
9
```

The console output shows the execution results:

```
groovy> package com.app
groovy> class GroovyDecisionMakingExample1 {
groovy>     static void main(args) {
groovy>         String s1='Javatpoint'
groovy>         println s1
groovy>         println 'This is tutorial on Groovy at' +s1
groovy>     }
groovy> }

Javatpoint
This is tutorial on Groovy atJavatpoint

Execution complete. Result was null. Elapsed time: 21ms.
```

The Windows taskbar at the bottom shows the search bar, task view button, and several application icons. The system tray on the right indicates the language is ENG US, the date is 17-02-2025, and the time is 12:45.

8. Groovy program to demonstrate string manipulation with concatenation using double quotes.

The screenshot shows the GroovyConsole application with a menu bar (File, Edit, View, History, Script, Help) and a toolbar. The script editor contains the following code:

```
1 package com.app
2 class GroovyDecisionMakingExample1 {
3     static void main(args) {
4         String s1="Javatpoint"
5         println s1
6         println "This is tutorial on Groovy at" +s1
7     }
8 }
9
```

The console output shows the execution results:

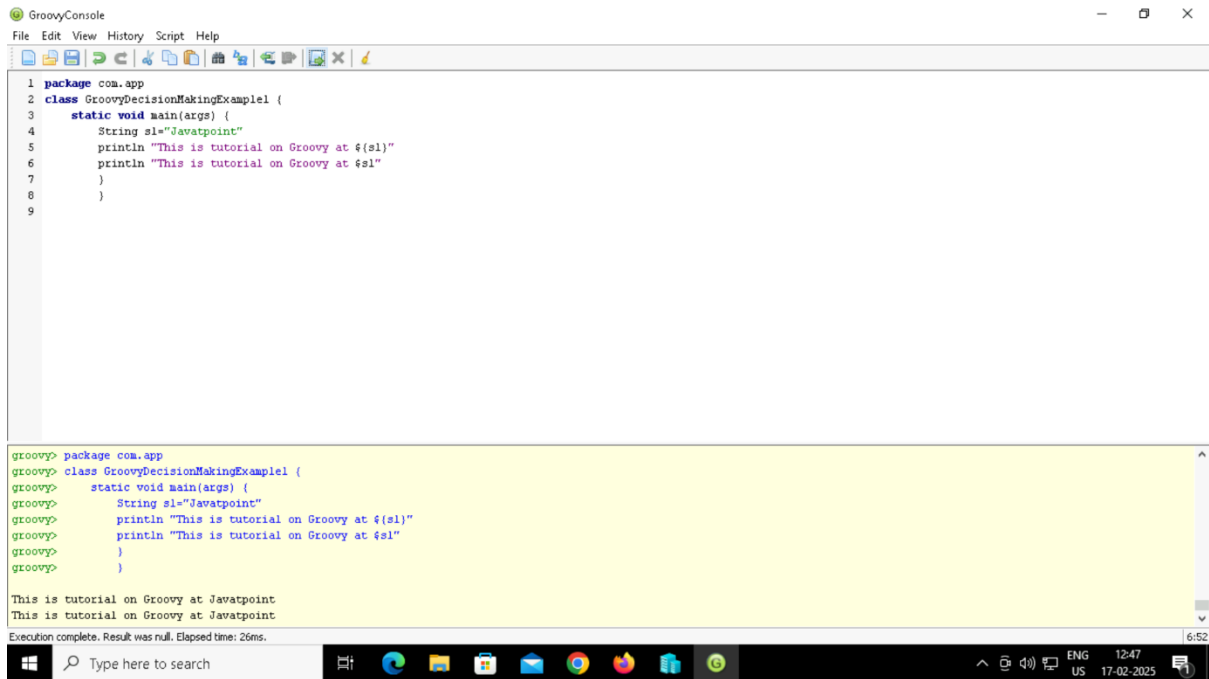
```
groovy> package com.app
groovy> class GroovyDecisionMakingExample1 {
groovy>     static void main(args) {
groovy>         String s1="Javatpoint"
groovy>         println s1
groovy>         println "This is tutorial on Groovy at" +s1
groovy>     }
groovy> }

Javatpoint
This is tutorial on Groovy atJavatpoint

Execution complete. Result was null. Elapsed time: 23ms.
```

The Windows taskbar at the bottom shows the search bar, task view button, and several application icons. The system tray on the right indicates the language is ENG US, the date is 17-02-2025, and the time is 12:45.

9. Groovy program to demonstrate string interpolation with both \${} and \$.



The screenshot shows the GroovyConsole application with a menu bar (File, Edit, View, History, Script, Help) and a toolbar. The script editor contains the following code:

```
1 package com.app
2 class GroovyDecisionMakingExample1 {
3     static void main(args) {
4         String s1="Javatpoint"
5         println "This is tutorial on Groovy at ${s1}"
6         println "This is tutorial on Groovy at ${s1}"
7     }
8 }
9
```

The console output shows the execution of the script:

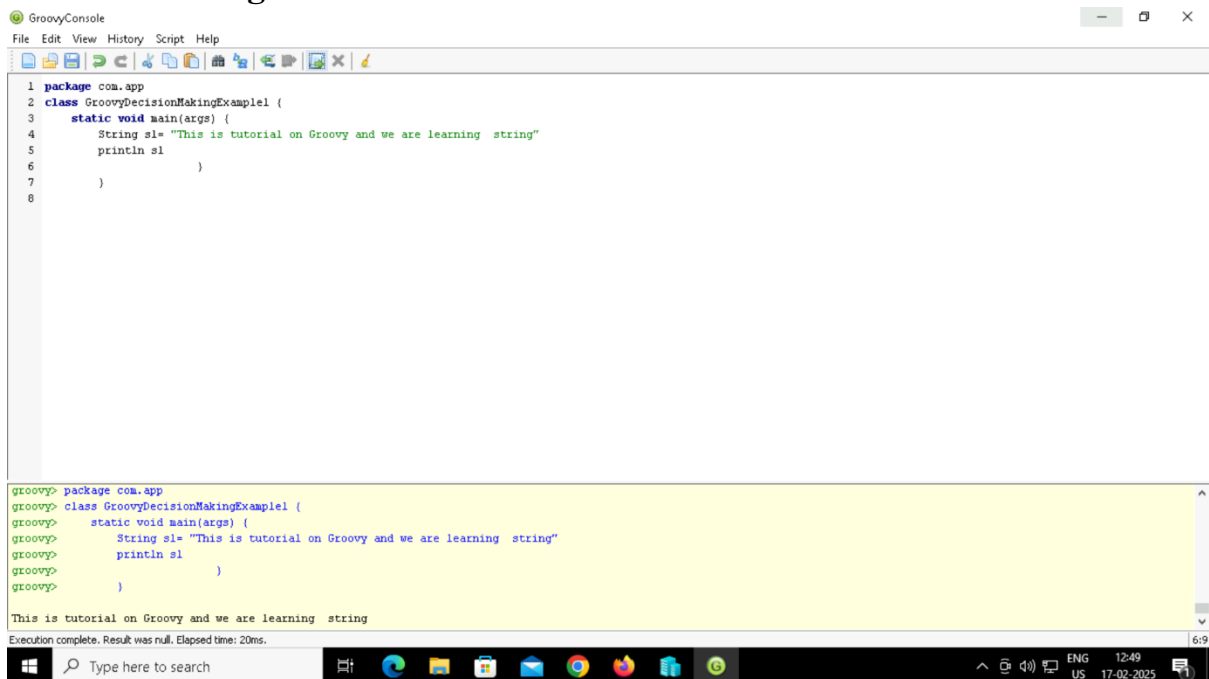
```
groovy> package com.app
groovy> class GroovyDecisionMakingExample1 {
groovy>     static void main(args) {
groovy>         String s1="Javatpoint"
groovy>         println "This is tutorial on Groovy at ${s1}"
groovy>         println "This is tutorial on Groovy at ${s1}"
groovy>     }
groovy> }

This is tutorial on Groovy at Javatpoint
This is tutorial on Groovy at Javatpoint

Execution complete. Result was null. Elapsed time: 26ms.
```

The Windows taskbar at the bottom shows the search bar, task view button, and several application icons. The system tray on the right indicates the language is ENG US, the date is 17-02-2025, and the time is 12:47.

10. Groovy program to demonstrate use of triple single quotes (') for multi-line strings.



The screenshot shows the GroovyConsole application with a menu bar (File, Edit, View, History, Script, Help) and a toolbar. The script editor contains the following code:

```
1 package com.app
2 class GroovyDecisionMakingExample1 {
3     static void main(args) {
4         String s1= "This is tutorial on Groovy and we are learning string"
5         println s1
6     }
7 }
8
```

The console output shows the execution of the script:

```
groovy> package com.app
groovy> class GroovyDecisionMakingExample1 {
groovy>     static void main(args) {
groovy>         String s1= "This is tutorial on Groovy and we are learning string"
groovy>         println s1
groovy>     }
groovy> }

This is tutorial on Groovy and we are learning string

Execution complete. Result was null. Elapsed time: 20ms.
```

The Windows taskbar at the bottom shows the search bar, task view button, and several application icons. The system tray on the right indicates the language is ENG US, the date is 17-02-2025, and the time is 12:49.

11. Groovy program using Triple-single-quoted strings for multiple lines in a single variable.

The screenshot shows the GroovyConsole application with a menu bar (File, Edit, View, History, Script, Help) and a toolbar. The main editor contains the following Groovy code:

```
1 package com.app
2 class GroovyStringExample5 {
3     static void main(args) {
4         String s1 = '''This is line 1
5                     This is line 2
6                     This is line 3
7                     This is line 4
8                     This is line 5'''
9         println s1
10    }
11 }
12
13
```

The console output area shows the execution results:

```
groovy> package com.app
groovy> class GroovyStringExample5 {
groovy>     static void main(args) {
groovy>         String s1 = '''This is line 1
groovy>                     This is line 2
groovy>                     This is line 3
groovy>                     This is line 4
groovy>                     This is line 5'''
groovy>         println s1
groovy>     }
groovy> }

This is line 1
      This is line 2
      This is line 3
      This is line 4
      This is line 5
```

The status bar at the bottom indicates "Execution complete. Result was null. Elapsed time: 21ms." and the system clock shows 12:51 on 17-02-2025.

12. Groovy program to demonstrate triple double quotes (""") for defining multi-line strings .

The screenshot shows the GroovyConsole application with a menu bar (File, Edit, View, History, Script, Help) and a toolbar. The main editor contains the following Groovy code:

```
1 package com.app
2 class GroovyStringExample6 {
3     static void main(args) {
4         String s1 = """This is groovy tutorial and we are learning string"""
5         println s1
6     }
7 }
8
9
```

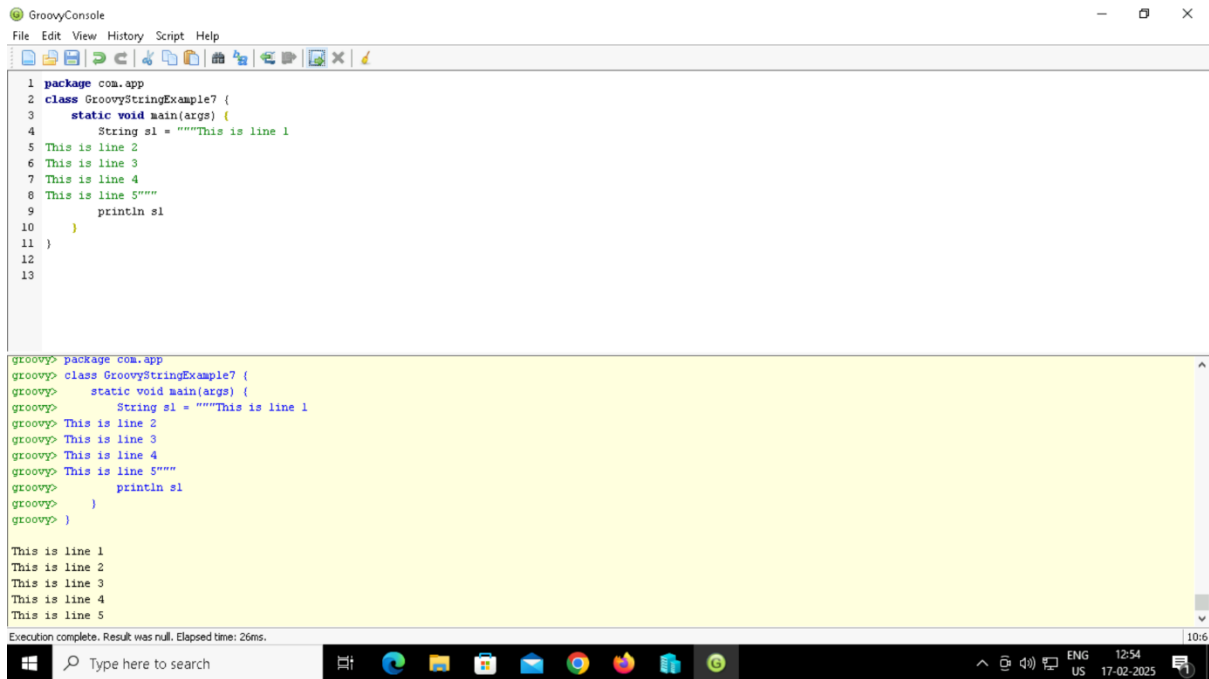
The console output area shows the execution results:

```
groovy> package com.app
groovy> class GroovyStringExample6 {
groovy>     static void main(args) {
groovy>         String s1 = """This is groovy tutorial and we are learning string"""
groovy>         println s1
groovy>     }
groovy> }

This is groovy tutorial and we are learning string
```

The status bar at the bottom indicates "Execution complete. Result was null. Elapsed time: 28ms." and the system clock shows 12:53 on 17-02-2025.

13. Groovy program that defines a multi-line string using triple double quotes (""") and prints the string .



The screenshot shows the GroovyConsole application with a menu bar (File, Edit, View, History, Script, Help) and a toolbar. The editor contains a Groovy script:

```
1 package com.app
2 class GroovyStringExample7 {
3     static void main(args) {
4         String s1 = """This is line 1
5 This is line 2
6 This is line 3
7 This is line 4
8 This is line 5"""
9         println s1
10    }
11 }
12
13
```

The console output shows the execution of the script:

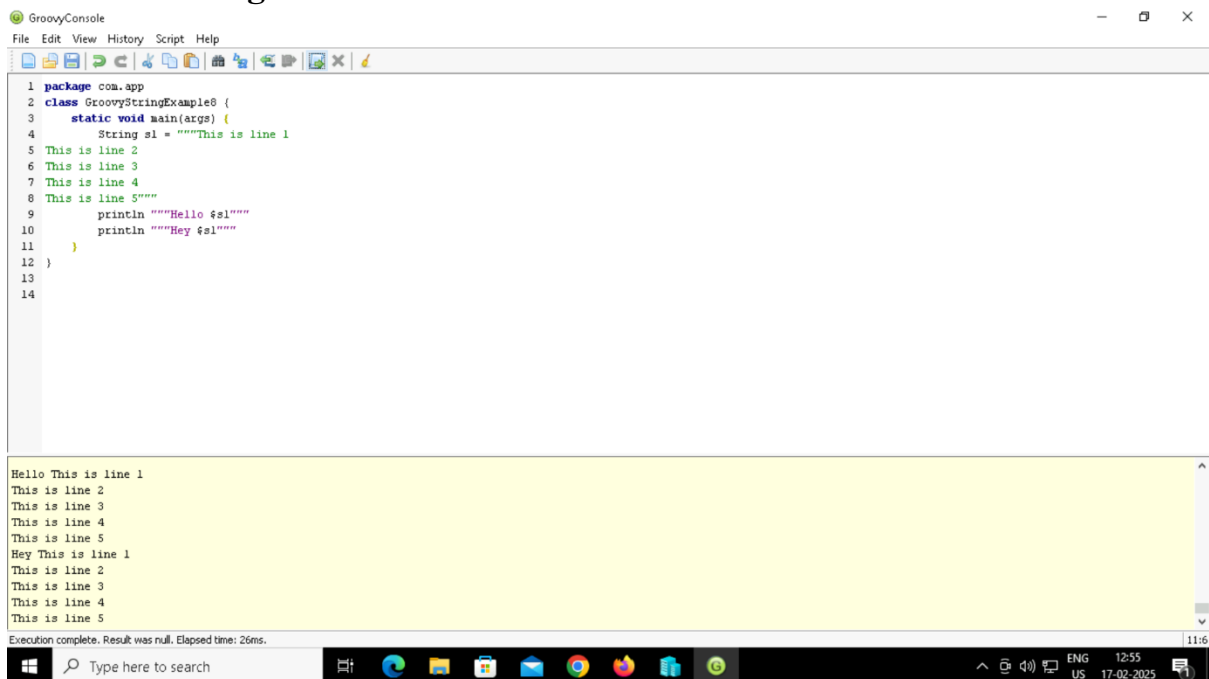
```
groovy> package com.app
groovy> class GroovyStringExample7 {
groovy>     static void main(args) {
groovy>         String s1 = """This is line 1
groovy> This is line 2
groovy> This is line 3
groovy> This is line 4
groovy> This is line 5"""
groovy>         println s1
groovy>     }
groovy> }
groovy>

This is line 1
This is line 2
This is line 3
This is line 4
This is line 5

Execution complete. Result was null. Elapsed time: 26ms.
```

The Windows taskbar at the bottom shows the search bar and various application icons.

14. Groovy program demonstrates the use of string interpolation with multi-line strings



The screenshot shows the GroovyConsole application with a menu bar (File, Edit, View, History, Script, Help) and a toolbar. The editor contains a Groovy script:

```
1 package com.app
2 class GroovyStringExample8 {
3     static void main(args) {
4         String s1 = """This is line 1
5 This is line 2
6 This is line 3
7 This is line 4
8 This is line 5"""
9         println """Hello $s1"""
10        println """Hey $s1"""
11    }
12 }
13
14
```

The console output shows the execution of the script:

```
Hello This is line 1
This is line 2
This is line 3
This is line 4
This is line 5
Hey This is line 1
This is line 2
This is line 3
This is line 4
This is line 5

Execution complete. Result was null. Elapsed time: 26ms.
```

The Windows taskbar at the bottom shows the search bar and various application icons.

15. Groovy program demonstrates the use of slashy strings (/) to define string.

The screenshot shows the GroovyConsole application with a menu bar (File, Edit, View, History, Script, Help) and a toolbar. The script editor contains the following code:

```
1 package com.app
2 class GroovyStringExample9 {
3     static void main(args) {
4         String s1 = /This is groovy tutorial and we are learning string/
5         println s1
6     }
7 }
8
9
```

The console output shows the execution of the script:

```
groovy> package com.app
groovy> class GroovyStringExample9 {
groovy>     static void main(args) {
groovy>         String s1 = /This is groovy tutorial and we are learning string/
groovy>         println s1
groovy>     }
groovy> }

This is groovy tutorial and we are learning string

Execution complete. Result was null. Elapsed time: 25ms.
```

The Windows taskbar at the bottom shows the search bar and various application icons.

16. Groovy program demonstrates the use of string interpolation with multi-line strings

The screenshot shows the GroovyConsole application with a menu bar (File, Edit, View, History, Script, Help) and a toolbar. The script editor contains the following code:

```
1 package com.app
2 class GroovyStringExample11 {
3     static void main(args) {
4         String s1 = '''This is line 1
5 This is line 2
6 This is line 3
7 This is line 4
8 This is line 5'''
9         println s1
10    }
11 }
```

The console output shows the execution of the script:

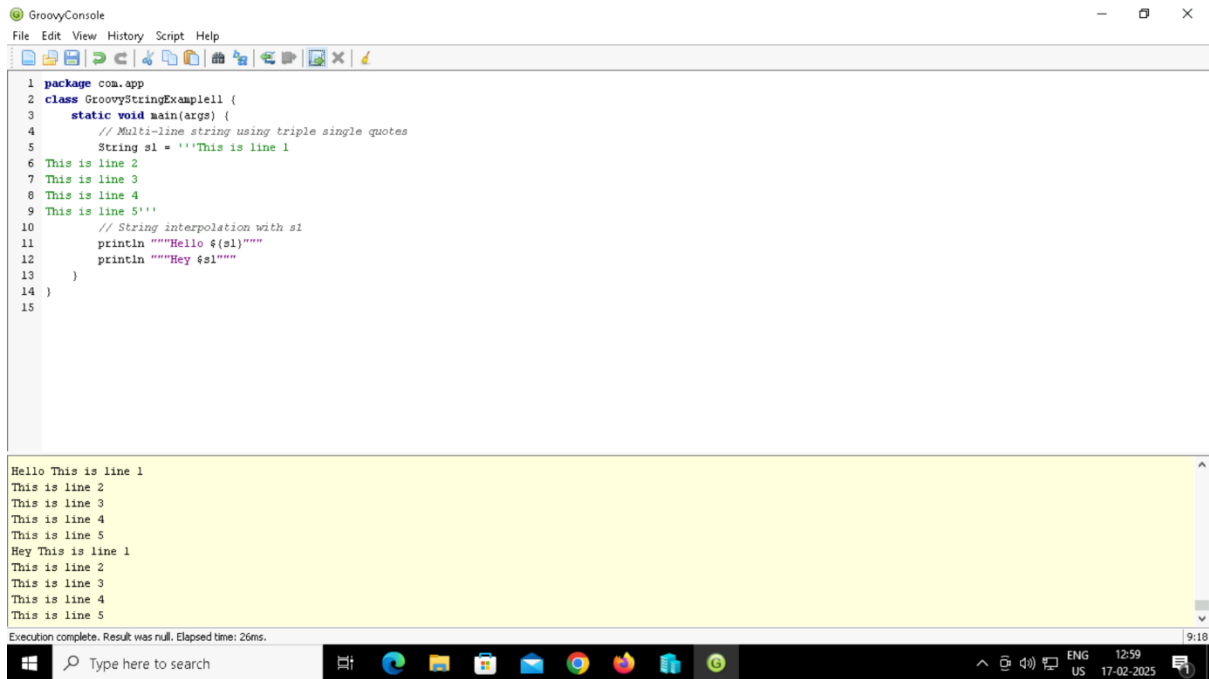
```
groovy> package com.app
groovy> class GroovyStringExample11 {
groovy>     static void main(args) {
groovy>         String s1 = '''This is line 1
groovy> This is line 2
groovy> This is line 3
groovy> This is line 4
groovy> This is line 5'''
groovy>         println s1
groovy>     }
groovy> }

This is line 1
This is line 2
This is line 3
This is line 4
This is line 5

Execution complete. Result was null. Elapsed time: 22ms.
```

The Windows taskbar at the bottom shows the search bar and various application icons.

17. Groovy program demonstrates multi-line strings using triple single quotes.



The screenshot shows the GroovyConsole application with a menu bar (File, Edit, View, History, Script, Help) and a toolbar. The script editor contains the following code:

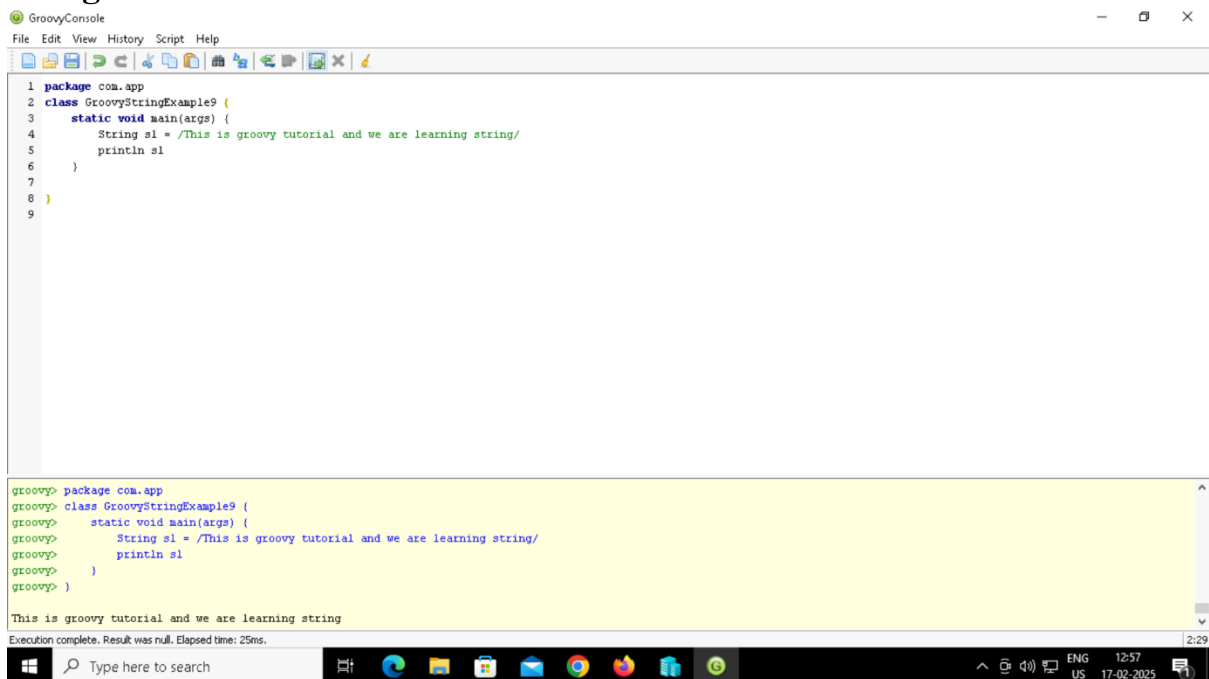
```
1 package com.app
2 class GroovyStringExample11 {
3     static void main(args) {
4         // Multi-line string using triple single quotes
5         String s1 = '''This is line 1
6 This is line 2
7 This is line 3
8 This is line 4
9 This is line 5'''
10
11         // String interpolation with s1
12         println """Hello ${s1}"""
13         println """Hey $s1"""
14     }
15 }
```

The output console shows the following output:

```
Hello This is line 1
This is line 2
This is line 3
This is line 4
This is line 5
Hey This is line 1
This is line 2
This is line 3
This is line 4
This is line 5
```

Execution complete. Result was null. Elapsed time: 26ms.

18. Groovy program demonstrates the use of slashy strings (/) to define string.



The screenshot shows the GroovyConsole application with a menu bar (File, Edit, View, History, Script, Help) and a toolbar. The script editor contains the following code:

```
1 package com.app
2 class GroovyStringExample9 {
3     static void main(args) {
4         String s1 = /This is groovy tutorial and we are learning string/
5         println s1
6     }
7 }
8
9
```

The output console shows the following output:

```
groovy> package com.app
groovy> class GroovyStringExample9 {
groovy>     static void main(args) {
groovy>         String s1 = /This is groovy tutorial and we are learning string/
groovy>         println s1
groovy>     }
groovy> }
```

This is groovy tutorial and we are learning string

Execution complete. Result was null. Elapsed time: 25ms.

19. Groovy program demonstrates the use of dollar slashy strings (/) to define string.

The screenshot shows the GroovyConsole application with a menu bar (File, Edit, View, History, Script, Help) and a toolbar. The script editor contains the following code:

```
1 package com.app
2 class GroovyStringExample9 {
3     static void main(args) {
4         // Correct usage of dollar slashy strings
5         String s1 = $/This is groovy tutorial and we are learning string/
6         println s1
7     }
8 }
9
10
```

The console output area shows the execution of the script:

```
groovy> package com.app
groovy> class GroovyStringExample9 {
groovy>     static void main(args) {
groovy>         // Correct usage of dollar slashy strings
groovy>         String s1 = $/This is groovy tutorial and we are learning string/
groovy>         println s1
groovy>     }
groovy> }

This is groovy tutorial and we are learning string

Execution complete. Result was null. Elapsed time: 22ms.
```

The Windows taskbar at the bottom shows the search bar, taskbar icons, and system tray with the date 17-02-2025 and time 13:01.

20. Groovy program using GString notation with \$ and / to define a multi-line string

The screenshot shows the GroovyConsole application with a menu bar (File, Edit, View, History, Script, Help) and a toolbar. The script editor contains the following code:

```
1 package com.app
2 class GroovyStringExample13 {
3     static void main(args) {
4         // This defines a multi-line string
5         String s1 = $/This is line 1
6         This is line 2
7         This is line 3
8         This is line 4
9         This is line 5/$
10        // Print the string
11        println s1
12    }
13 }
14
```

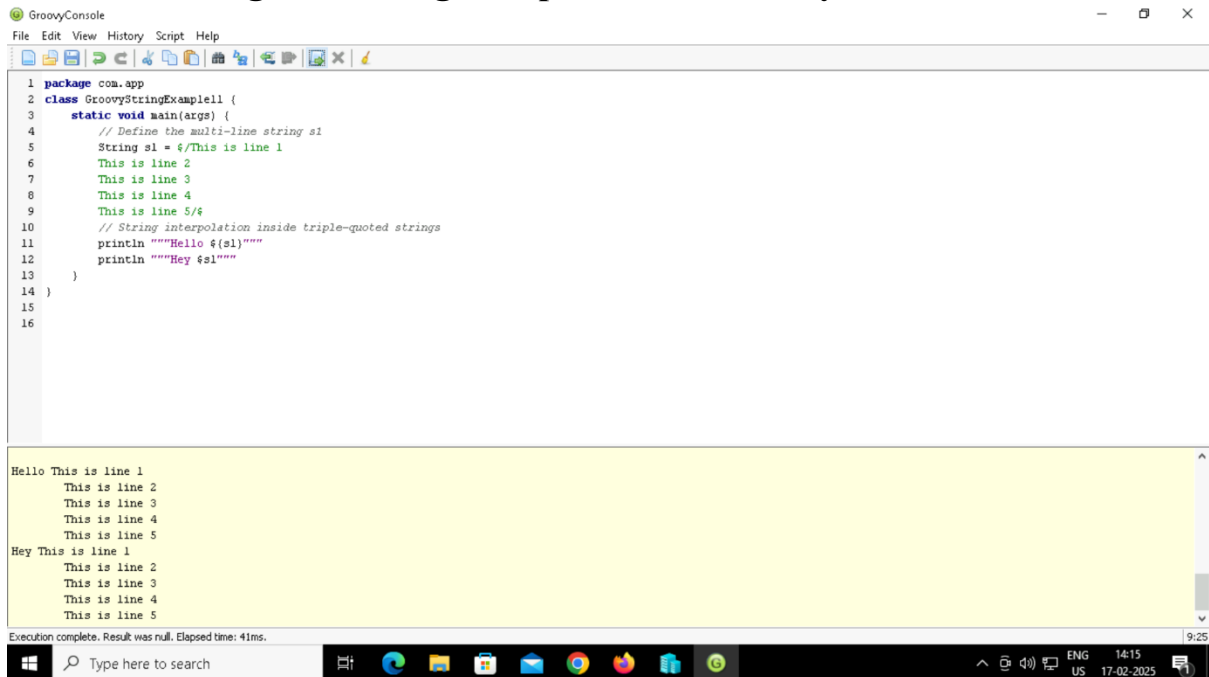
The console output area shows the execution of the script:

```
This is line 1
This is line 2
This is line 3
This is line 4
This is line 5

Execution complete. Result was null. Elapsed time: 536ms.
```

The Windows taskbar at the bottom shows the search bar, taskbar icons, and system tray with the date 17-02-2025 and time 14:13.

21. GString (\$ notation) and the triple-quoted string (""") to combine multi-line strings and string interpolation in Groovy.



The screenshot shows the GroovyConsole application window. The title bar reads "GroovyConsole". The menu bar includes "File", "Edit", "View", "History", "Script", and "Help". The toolbar contains icons for file operations and execution. The main text area contains the following Groovy code:

```
1 package com.app
2 class GroovyStringExample1 {
3     static void main(args) {
4         // Define the multi-line string s1
5         String s1 = $/This is line 1
6             This is line 2
7             This is line 3
8             This is line 4
9             This is line 5/$
10        // String interpolation inside triple-quoted strings
11        println ""Hello ${s1}""
12        println ""Hey ${s1}""
13    }
14 }
15
16
```

The output area at the bottom shows the results of the execution:

```
Hello This is line 1
    This is line 2
    This is line 3
    This is line 4
    This is line 5
Hey This is line 1
    This is line 2
    This is line 3
    This is line 4
    This is line 5
```

At the bottom of the window, a status bar indicates "Execution complete. Result was null. Elapsed time: 41ms." The Windows taskbar is visible at the very bottom, showing the search bar and various application icons.