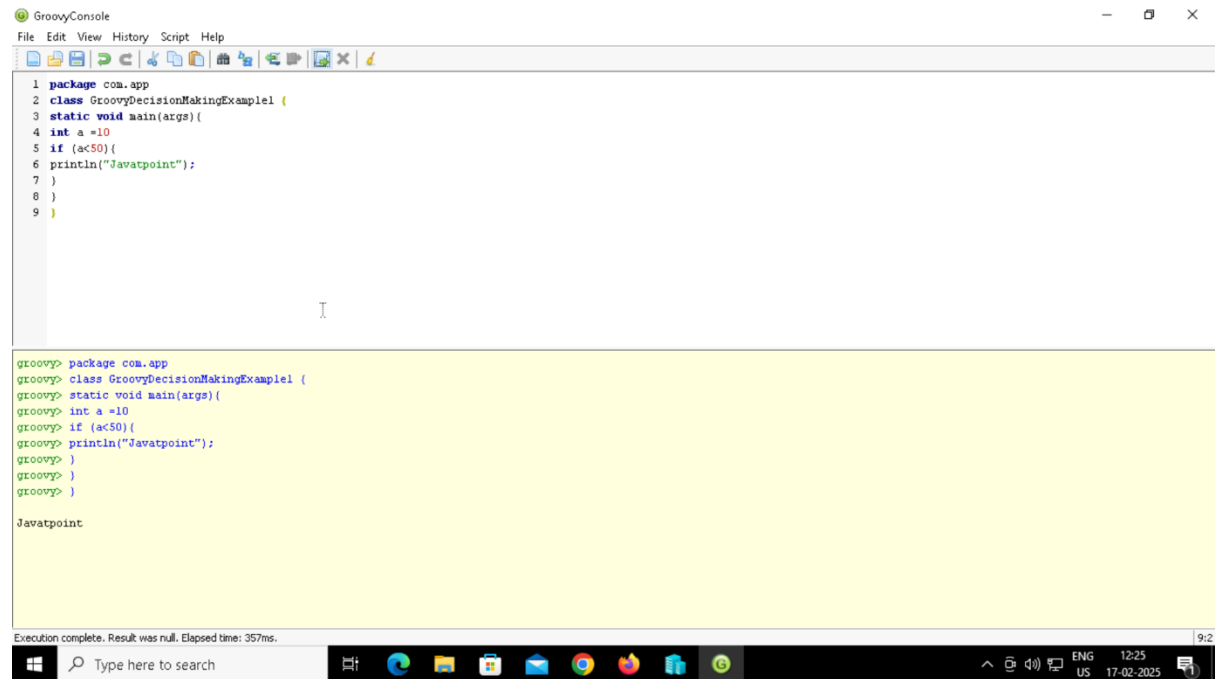


Groovy Project

If Statement

Example 1



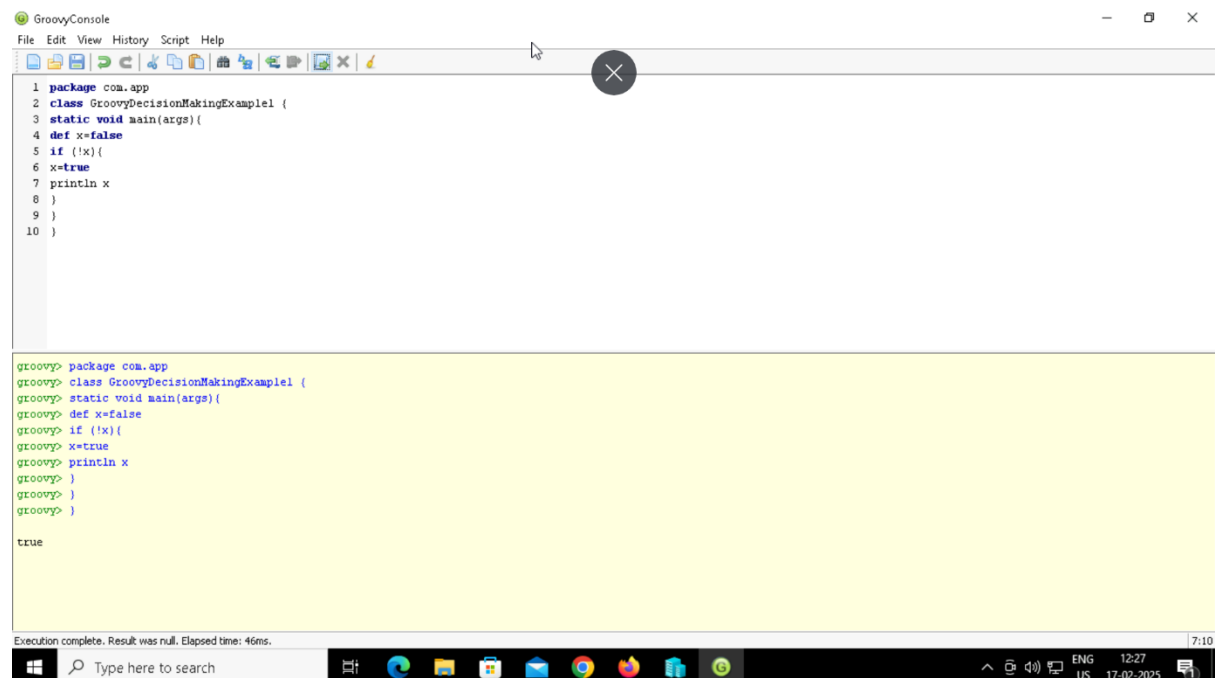
```
1 package com.app
2 class GroovyDecisionMakingExample1 {
3     static void main(args) {
4         int a = 10
5         if (a < 50) {
6             println("Javatpoint");
7         }
8     }
9 }
```

```
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: 357ms.
```

Example 2



```
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 }
```

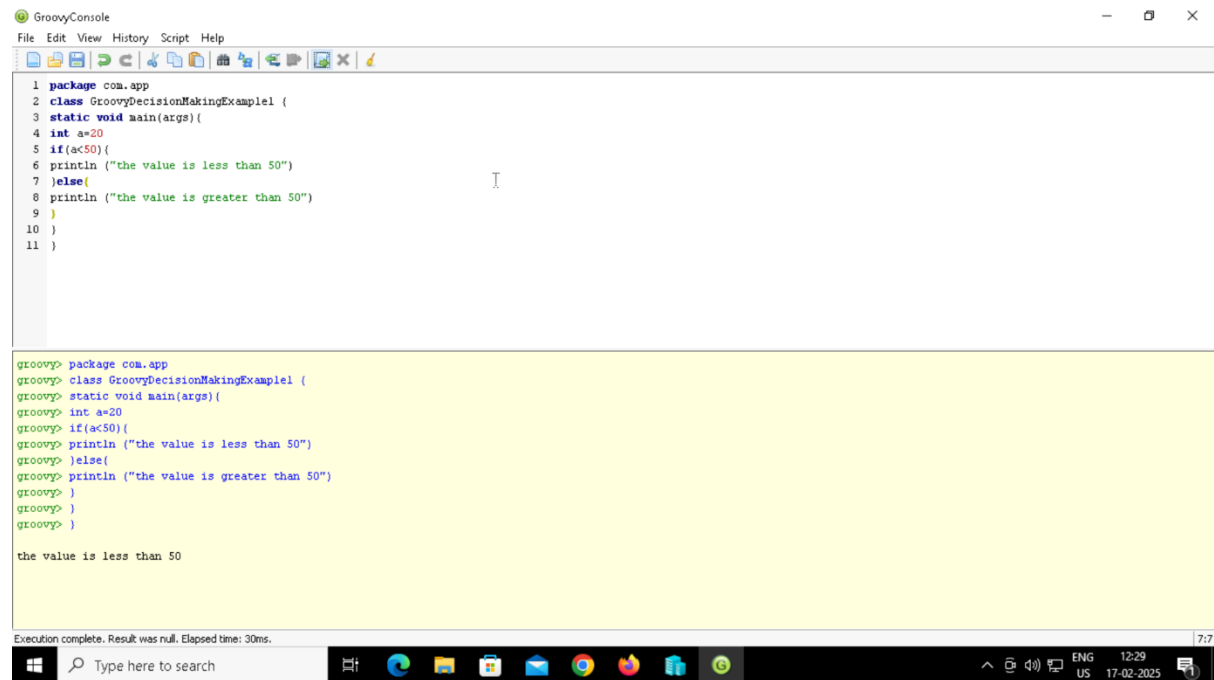
```
groovy> package com.app
groovy> class GroovyDecisionMakingExample1 {
groovy> static void main(args) {
groovy> def x = false
groovy> if (!x) {
groovy> x = true
groovy> println x
groovy> }
groovy> }
groovy> }

true

Execution complete. Result was null. Elapsed time: 46ms.
```

If else Statement

Example 3



The screenshot shows the GroovyConsole application with a menu bar (File, Edit, View, History, Script, Help) and a toolbar. The code editor contains the following Groovy 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         }else{
8             println ("the value is greater than 50")
9         }
10    }
11 }
```

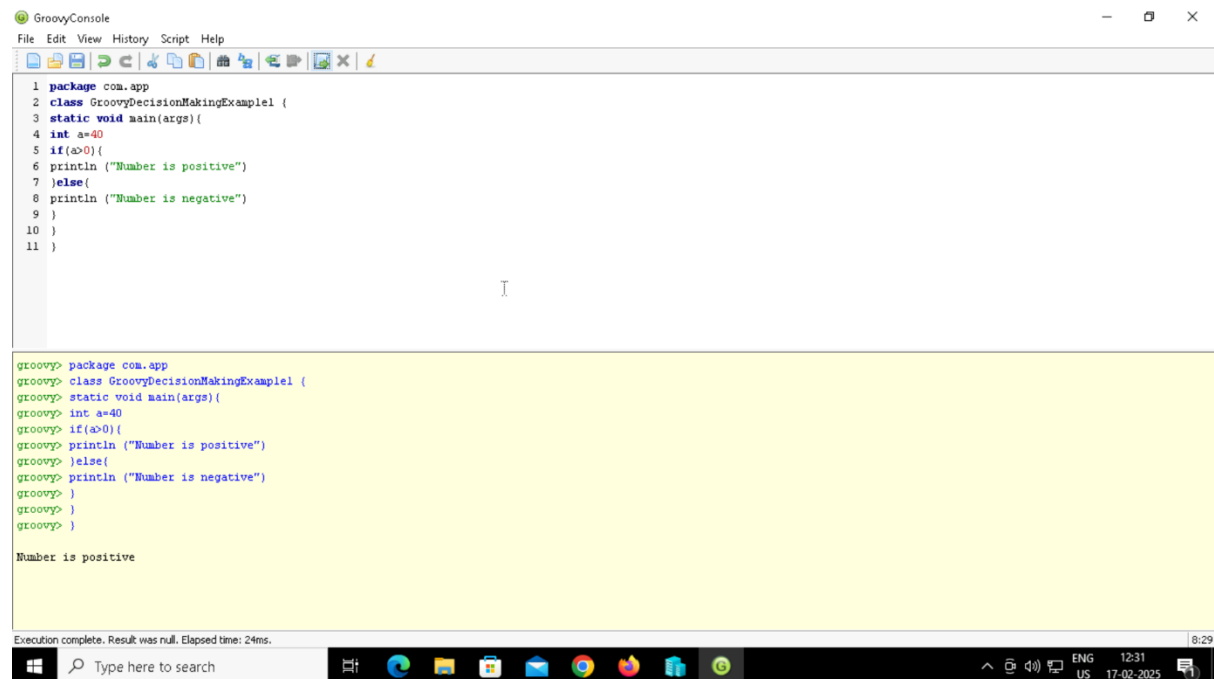
The output pane shows the execution result:

```
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> }else{
groovy> println ("the value is greater than 50")
groovy> }
groovy> }
groovy> }
groovy> }

the value is less than 50
```

At the bottom, a status bar indicates "Execution complete. Result was null. Elapsed time: 30ms." and a Windows taskbar is visible at the very bottom.

Example 4



The screenshot shows the GroovyConsole application with a menu bar (File, Edit, View, History, Script, Help) and a toolbar. The code editor contains the following Groovy 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         }else{
8             println ("Number is negative")
9         }
10    }
11 }
```

The output pane shows the execution result:

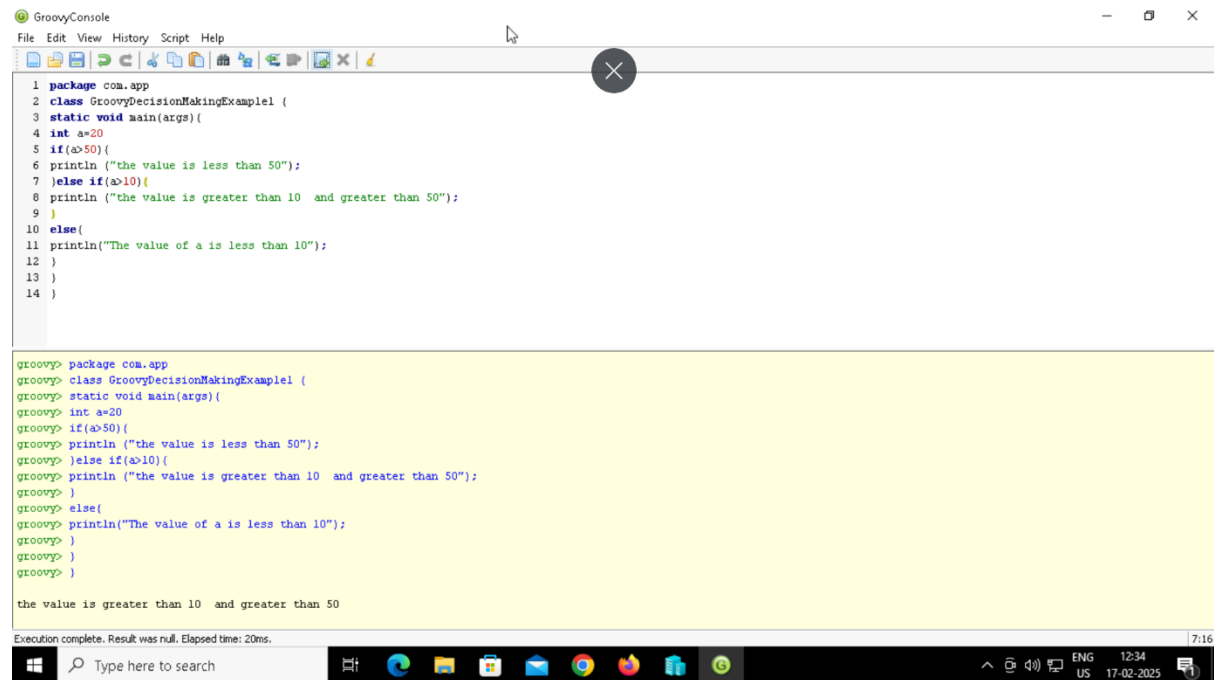
```
groovy> package com.app
groovy> class GroovyDecisionMakingExample1 {
groovy> static void main(args){
groovy> int a=40
groovy> if(a>0){
groovy> println ("Number is positive")
groovy> }else{
groovy> println ("Number is negative")
groovy> }
groovy> }
groovy> }
groovy> }

Number is positive
```

At the bottom, a status bar indicates "Execution complete. Result was null. Elapsed time: 24ms." and a Windows taskbar is visible at the very bottom.

Nested If Statement

Example 5



The screenshot shows the GroovyConsole application with a code editor and a console output area. The code defines a class `GroovyDecisionMakingExample1` with a `main` method. It uses a nested `if` statement to check the value of `a` (which is 20). The first `if` checks `a > 50`, which is false. The `else if` checks `a > 10`, which is true, and prints "the value is greater than 10 and greater than 50". The `else` block prints "The value of a is less than 10". The console output shows the result of the execution: "the value is greater than 10 and greater than 50".

```
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         } else if(a>10){
8             println ("the value is greater than 10 and greater than 50");
9         }
10        else{
11            println("The value of a is less than 10");
12        }
13    }
14 }
```

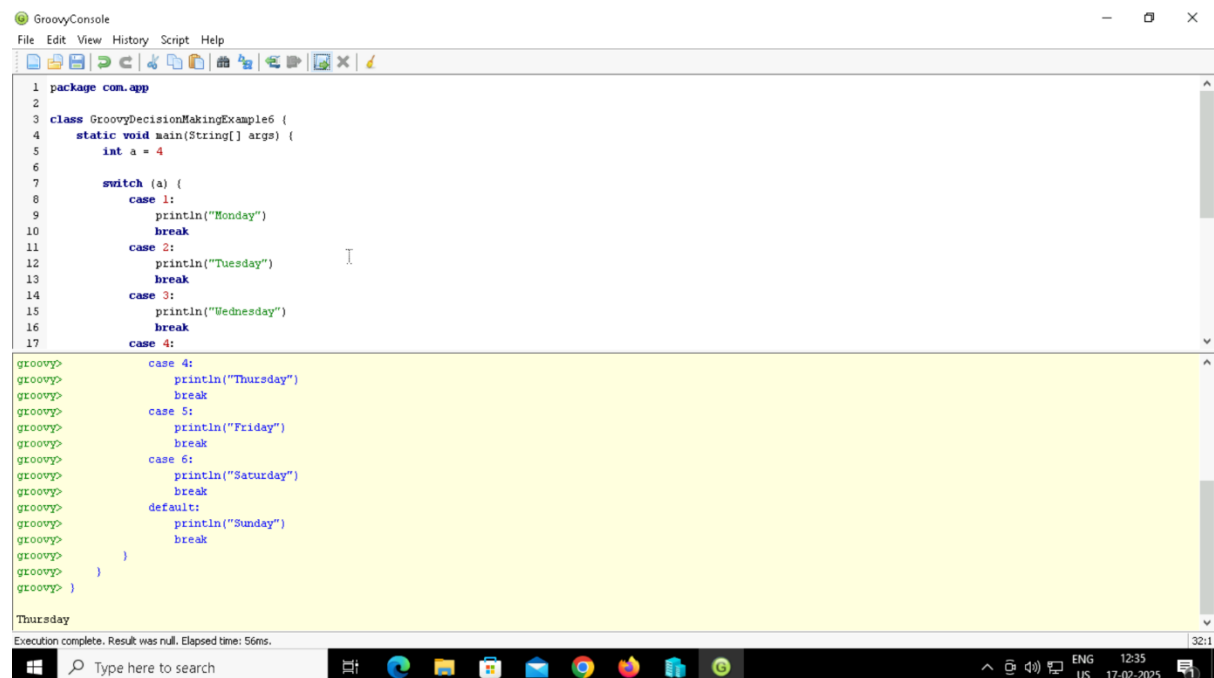
```
groovy> package com.app
groovy> class GroovyDecisionMakingExample1 {
groovy> static void main(args){
groovy> int a=20
groovy> if(a>50){
groovy> println ("the value is less than 50");
groovy> } else if(a>10){
groovy> println ("the value is greater than 10 and greater than 50");
groovy> }
groovy> else{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

the value is greater than 10 and greater than 50

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

Switch Statement

Example 6



The screenshot shows the GroovyConsole application with a code editor and a console output area. The code defines a class `GroovyDecisionMakingExample6` with a `main` method. It uses a `switch` statement to check the value of `a` (which is 4). The `switch` statement has cases for 1 through 6, each printing a day of the week and breaking. The `default` case prints "Sunday". The console output shows the result of the execution: "Thursday".

```
1 package com.app
2
3 class GroovyDecisionMakingExample6 {
4     static void main(String[] args) {
5         int a = 4
6
7         switch (a) {
8             case 1:
9                 println("Monday")
10                break
11             case 2:
12                 println("Tuesday")
13                break
14             case 3:
15                 println("Wednesday")
16                break
17             case 4:
18                 println("Thursday")
19                break
20             case 5:
21                 println("Friday")
22                break
23             case 6:
24                 println("Saturday")
25                break
26             default:
27                 println("Sunday")
28                break
29         }
30     }
31 }
```

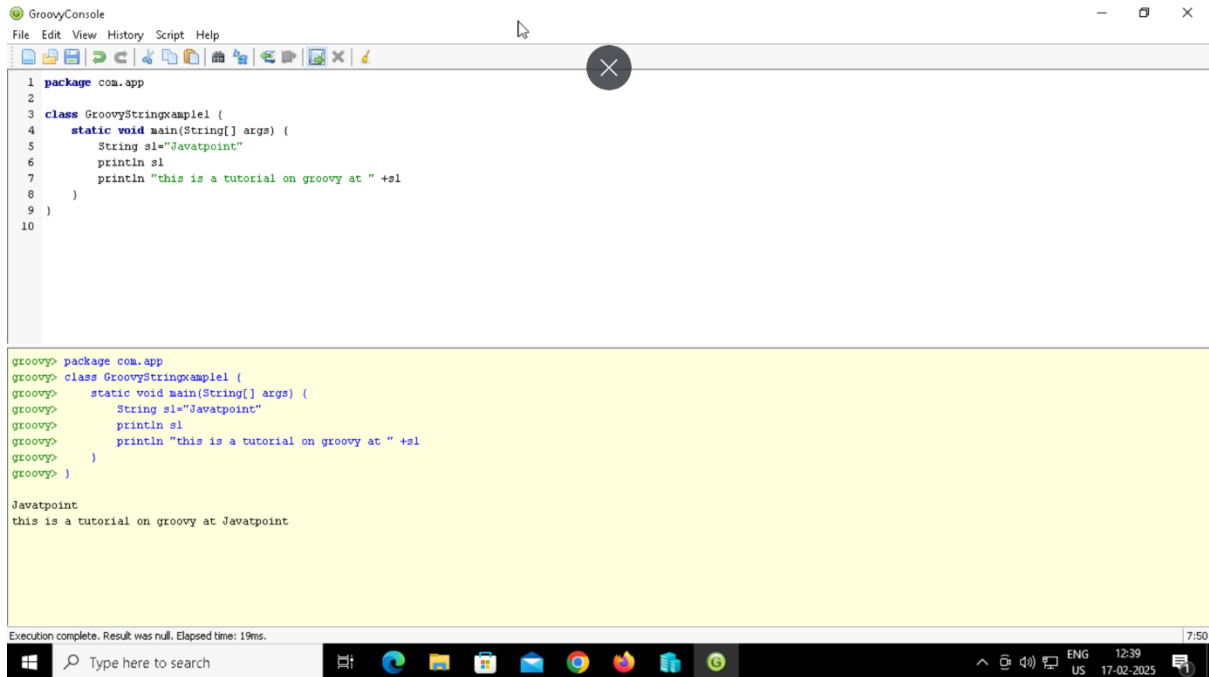
```
groovy> case 4:
groovy> println("Thursday")
groovy> break
groovy> case 5:
groovy> println("Friday")
groovy> break
groovy> case 6:
groovy> println("Saturday")
groovy> break
groovy> default:
groovy> println("Sunday")
groovy> break
groovy> }
groovy> }
```

Thursday

Execution complete. Result was null. Elapsed time: 56ms.

Single-quoted string

Example 1



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

```
1 package com.app
2
3 class GroovyStringxample1 {
4     static void main(String[] args) {
5         String s1="Javatpoint"
6         println s1
7         println "this is a tutorial on groovy at " +s1
8     }
9 }
10
```

The bottom pane shows the execution output:

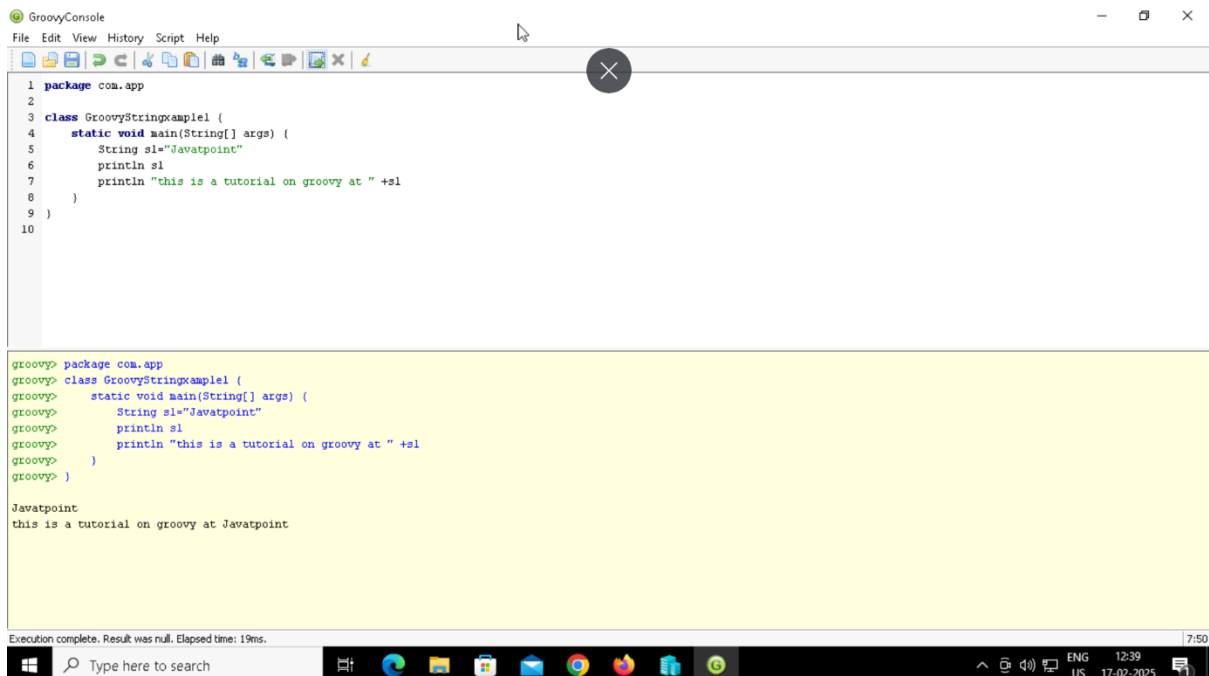
```
groovy> package com.app
groovy> class GroovyStringxample1 {
groovy>     static void main(String[] args) {
groovy>         String s1="Javatpoint"
groovy>         println s1
groovy>         println "this is a tutorial on groovy at " +s1
groovy>     }
groovy> }

Javatpoint
this is a tutorial on groovy at Javatpoint
```

At the bottom, a status bar indicates "Execution complete. Result was null. Elapsed time: 19ms." The Windows taskbar at the bottom shows the time as 12:39 on 17-02-2025.

Double-quoted string

Example 2



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

```
1 package com.app
2
3 class GroovyStringxample1 {
4     static void main(String[] args) {
5         String s1="Javatpoint"
6         println s1
7         println "this is a tutorial on groovy at " +s1
8     }
9 }
10
```

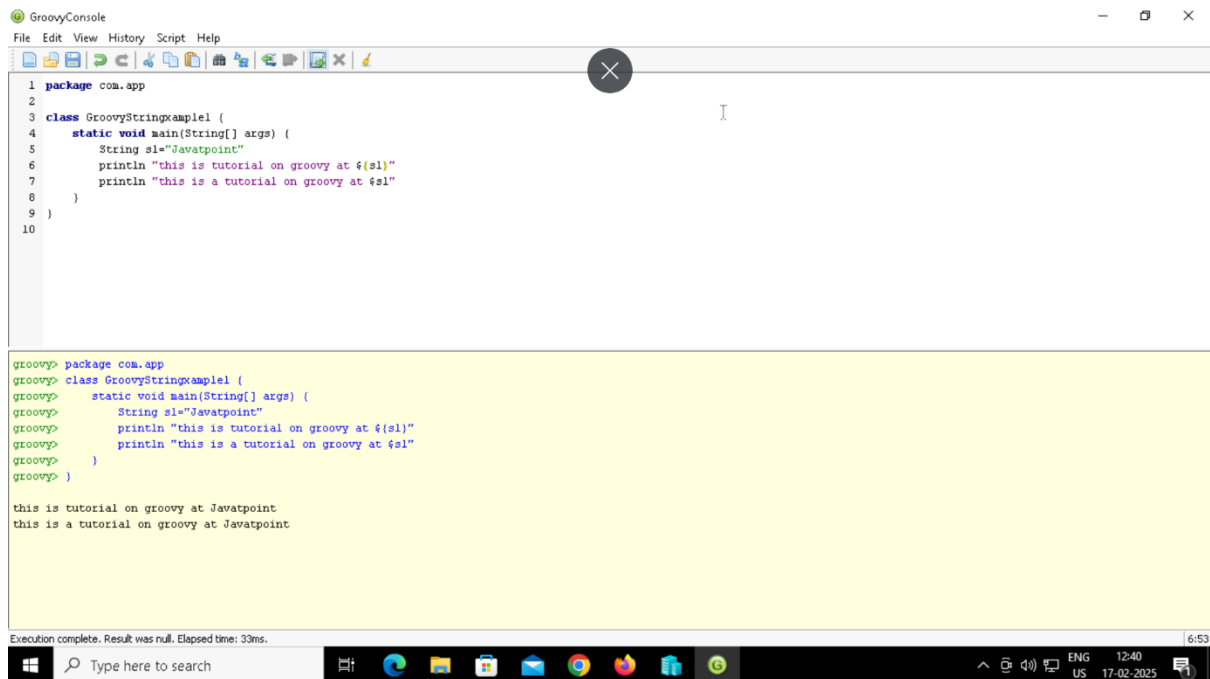
The bottom pane shows the execution output:

```
groovy> package com.app
groovy> class GroovyStringxample1 {
groovy>     static void main(String[] args) {
groovy>         String s1="Javatpoint"
groovy>         println s1
groovy>         println "this is a tutorial on groovy at " +s1
groovy>     }
groovy> }

Javatpoint
this is a tutorial on groovy at Javatpoint
```

At the bottom, a status bar indicates "Execution complete. Result was null. Elapsed time: 19ms." The Windows taskbar at the bottom shows the time as 12:39 on 17-02-2025.

Example 3



```
1 package com.app
2
3 class GroovyStringxample1 {
4     static void main(String[] args) {
5         String s1="Javatpoint"
6         println "this is tutorial on groovy at ${s1}"
7         println "this is a tutorial on groovy at ${s1}"
8     }
9 }
10
```

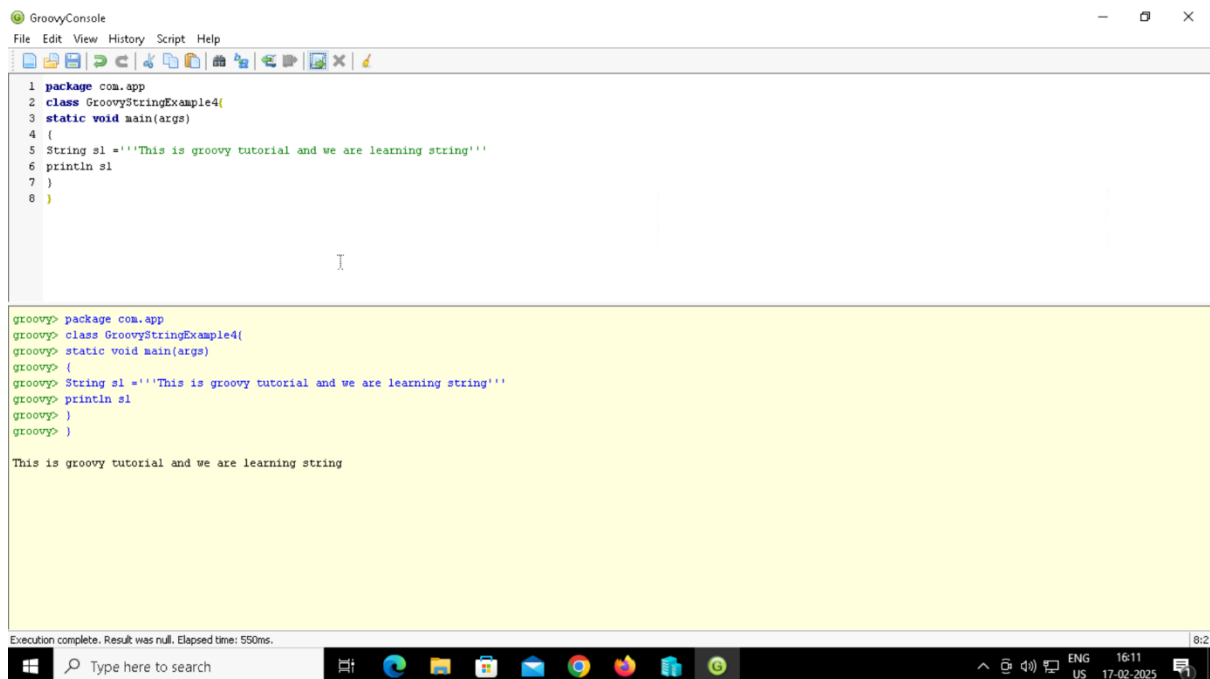
```
groovy> package com.app
groovy> class GroovyStringxample1 {
groovy>     static void main(String[] args) {
groovy>         String s1="Javatpoint"
groovy>         println "this is tutorial on groovy at ${s1}"
groovy>         println "this is a tutorial on groovy at ${s1}"
groovy>     }
groovy> }

this is tutorial on groovy at Javatpoint
this is a tutorial on groovy at Javatpoint
```

Execution complete. Result was null. Elapsed time: 33ms.

Triple -single-quoted string

Example 4



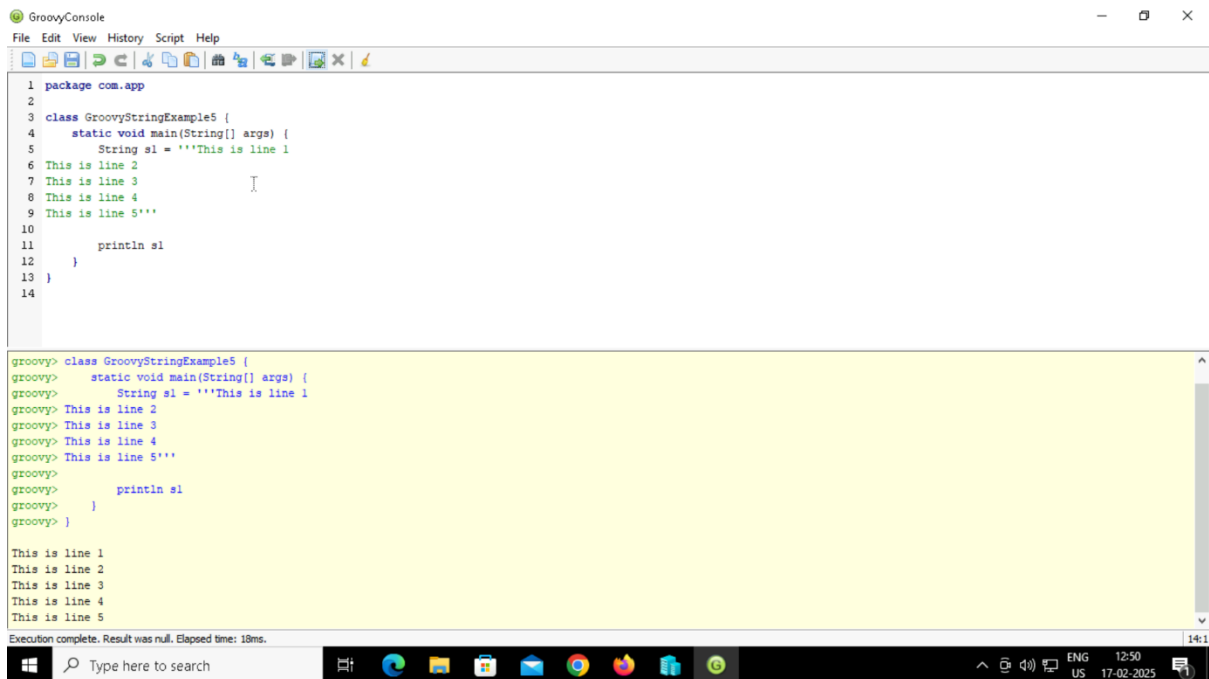
```
1 package com.app
2 class GroovyStringExample4{
3     static void main(args)
4     {
5         String s1 = '''This is groovy tutorial and we are learning string'''
6         println s1
7     }
8 }
```

```
groovy> package com.app
groovy> class GroovyStringExample4{
groovy> static void main(args)
groovy> {
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: 550ms.

Example 5



```
1 package com.app
2
3 class GroovyStringExample5 {
4     static void main(String[] args) {
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         println s1
12     }
13 }
14
```

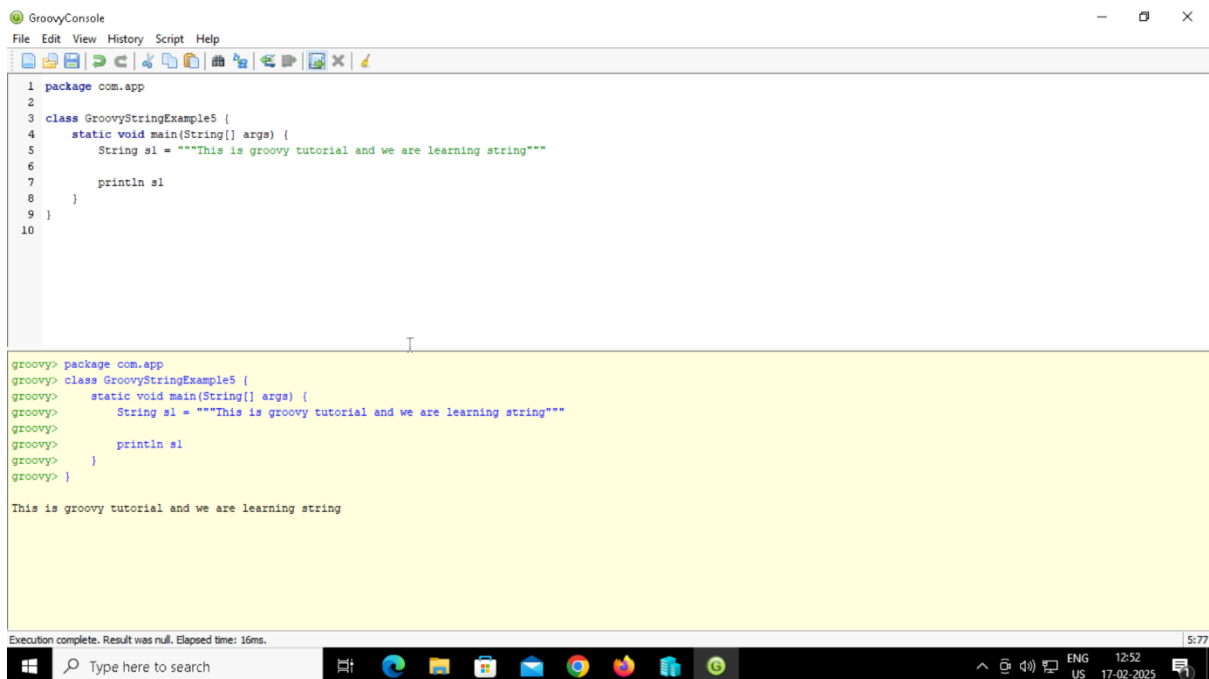
```
groovy> class GroovyStringExample5 {
groovy>     static void main(String[] args) {
groovy>         String s1 = '''This is line 1
groovy> This is line 2
groovy> This is line 3
groovy> This is line 4
groovy> This is line 5'''
groovy>
groovy>         println s1
groovy>     }
groovy> }

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: 18ms.

Triple-double-quoted string

Example 6



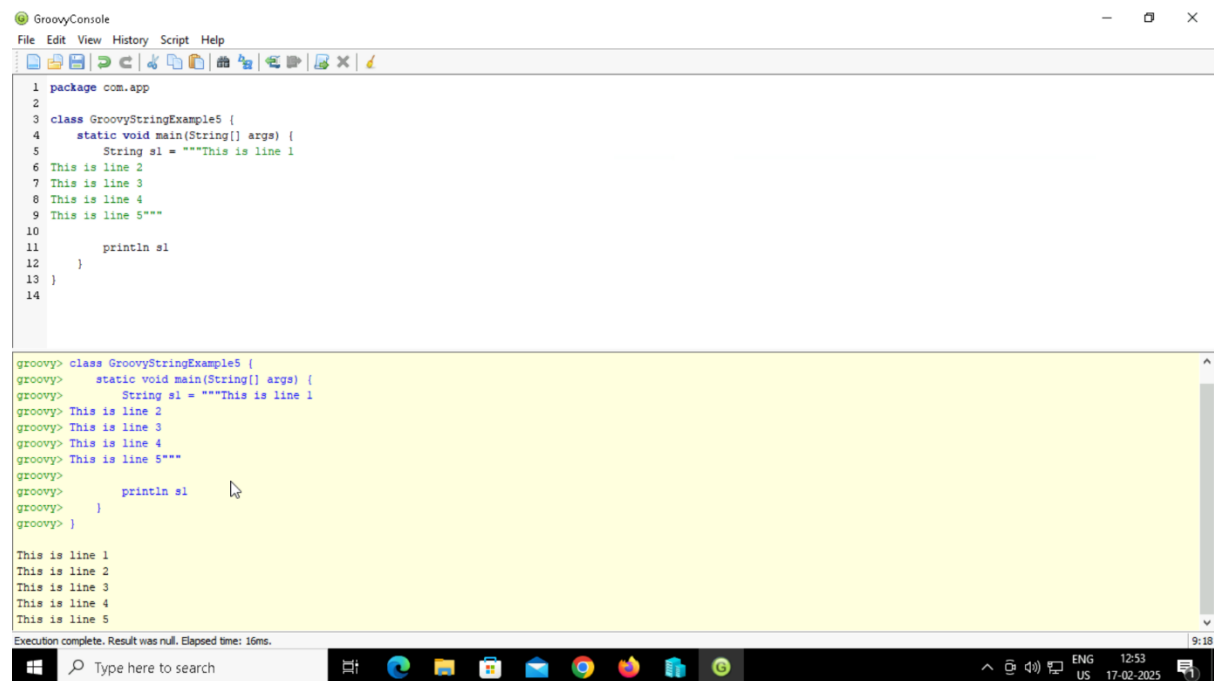
```
1 package com.app
2
3 class GroovyStringExample5 {
4     static void main(String[] args) {
5         String s1 = """This is groovy tutorial and we are learning string"""
6
7         println s1
8     }
9 }
10
```

```
groovy> package com.app
groovy> class GroovyStringExample5 {
groovy>     static void main(String[] args) {
groovy>         String s1 = """This is groovy tutorial and we are learning string"""
groovy>
groovy>         println s1
groovy>     }
groovy> }

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

Execution complete. Result was null. Elapsed time: 16ms.

Example 7



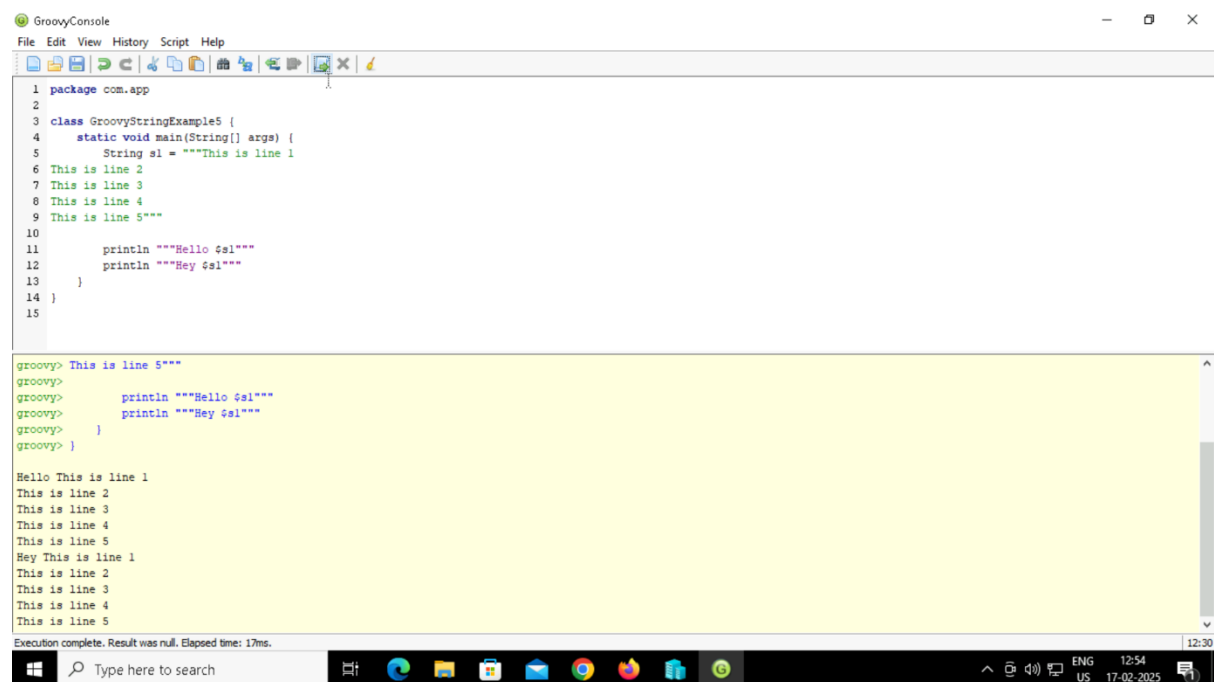
```
1 package com.app
2
3 class GroovyStringExample5 {
4     static void main(String[] args) {
5         String s1 = ""This is line 1
6         This is line 2
7         This is line 3
8         This is line 4
9         This is line 5""
10
11         println s1
12     }
13 }
14
```

```
groovy> class GroovyStringExample5 {
groovy>     static void main(String[] 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>
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: 16ms.
```

Example 8



```
1 package com.app
2
3 class GroovyStringExample5 {
4     static void main(String[] args) {
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         println ""Hello $s1""
12         println ""Hey $s1""
13     }
14 }
15
```

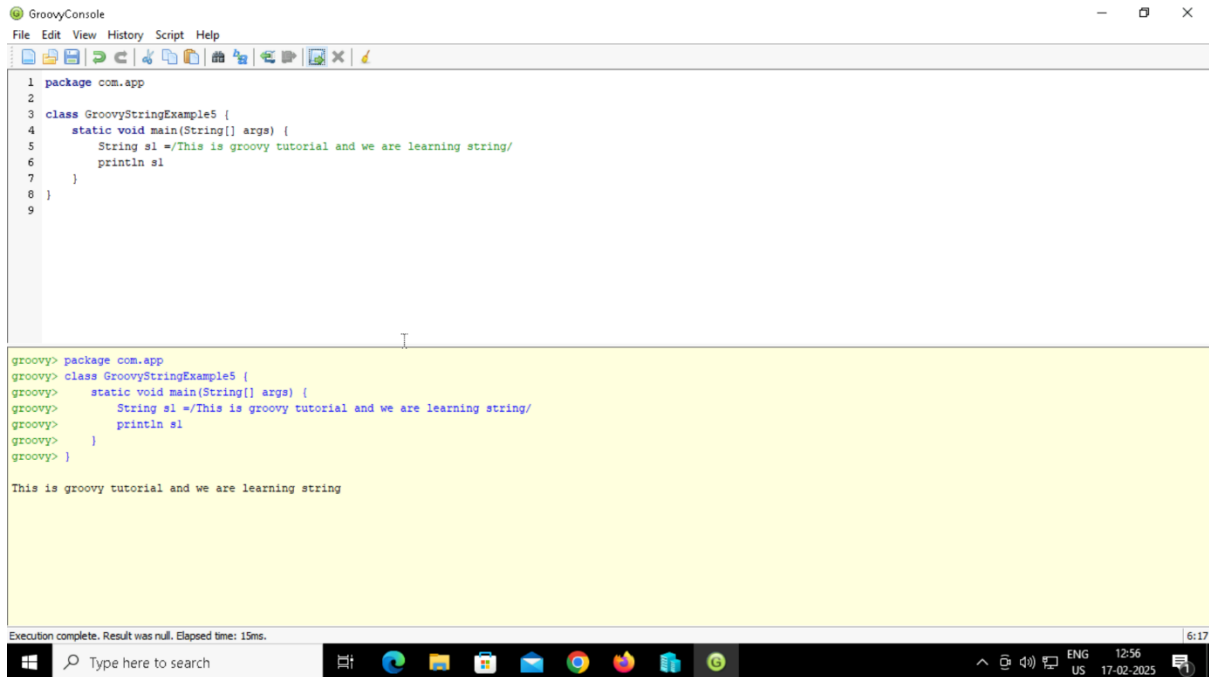
```
groovy> This is line 5""
groovy>
groovy>         println ""Hello $s1""
groovy>         println ""Hey $s1""
groovy>     }
groovy> }

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: 17ms.
```

Slasly string

Example 9



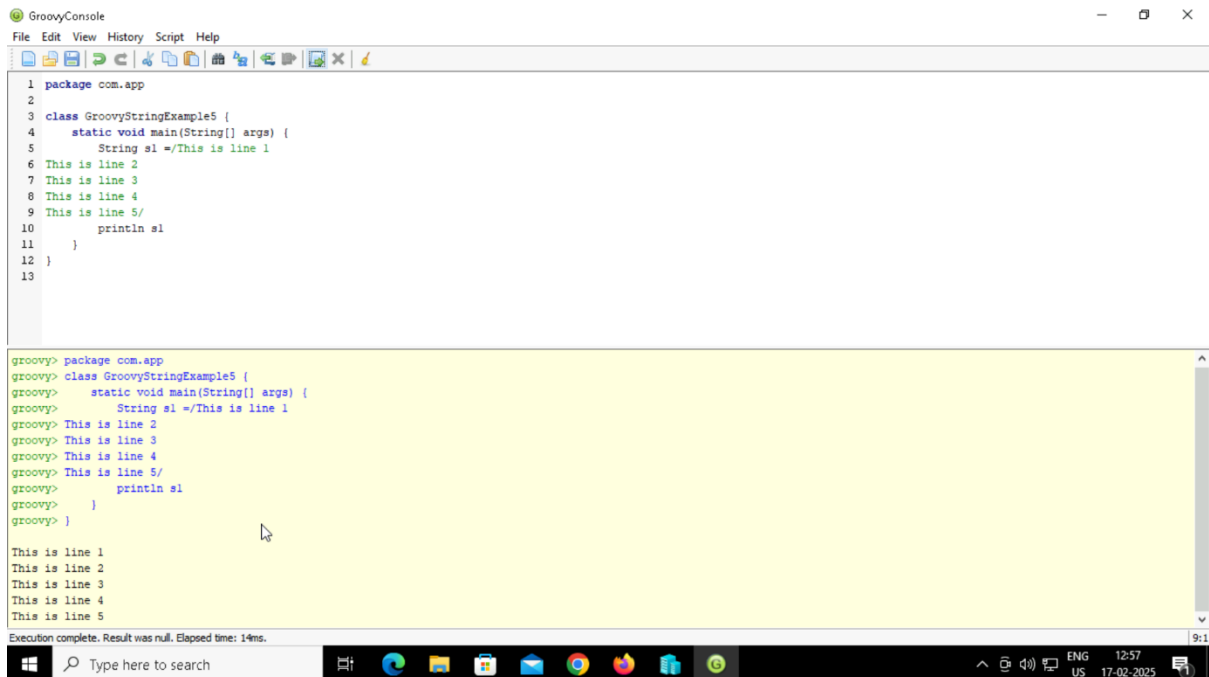
```
1 package com.app
2
3 class GroovyStringExample5 {
4     static void main(String[] args) {
5         String s1 =~/This is groovy tutorial and we are learning string/
6         println s1
7     }
8 }
9
```

```
groovy> package com.app
groovy> class GroovyStringExample5 {
groovy>     static void main(String[] 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: 15ms.

Example 10



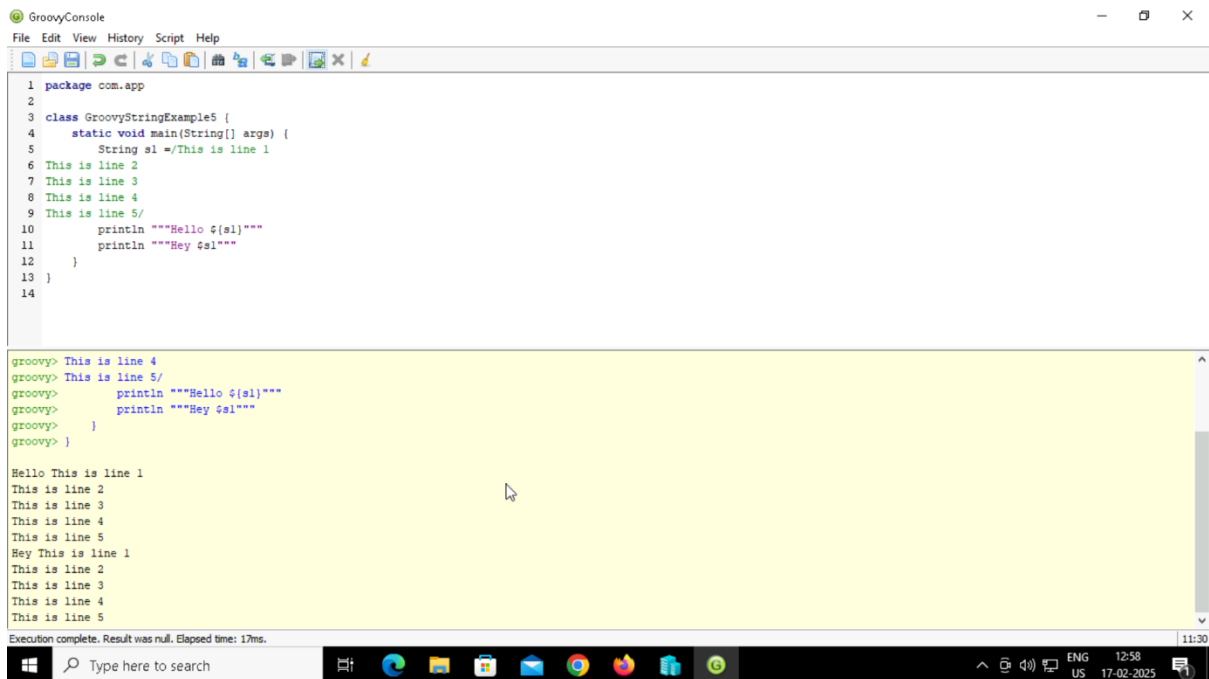
```
1 package com.app
2
3 class GroovyStringExample5 {
4     static void main(String[] args) {
5         String s1 =~/This is line 1
6         This is line 2
7         This is line 3
8         This is line 4
9         This is line 5/
10        println s1
11    }
12 }
13
```

```
groovy> package com.app
groovy> class GroovyStringExample5 {
groovy>     static void main(String[] 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: 14ms.

Example 11



```
1 package com.app
2
3 class GroovyStringExample5 {
4     static void main(String[] args) {
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        println ""Hello ${s1}""
11        println ""Hey ${s1}""
12    }
13 }
14
```

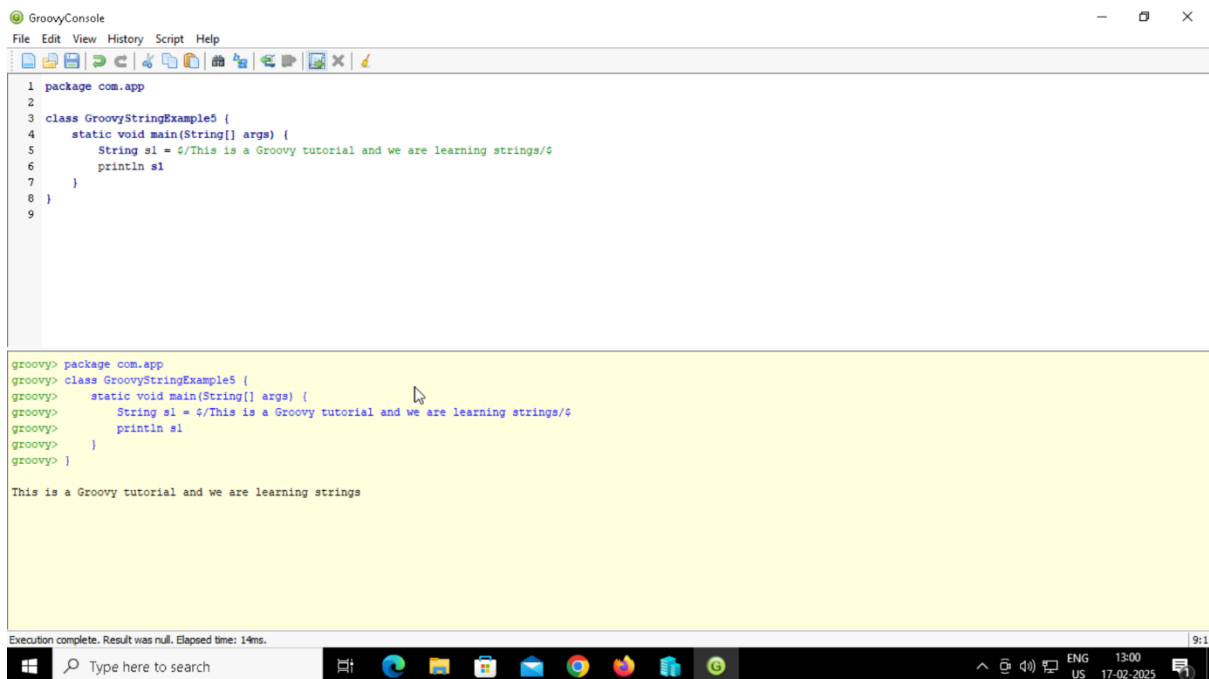
```
groovy> This is line 4
groovy> This is line 5/
groovy>        println ""Hello ${s1}""
groovy>        println ""Hey ${s1}""
groovy>    }
groovy> }

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: 17ms.

Dollar slashy string

Example 12



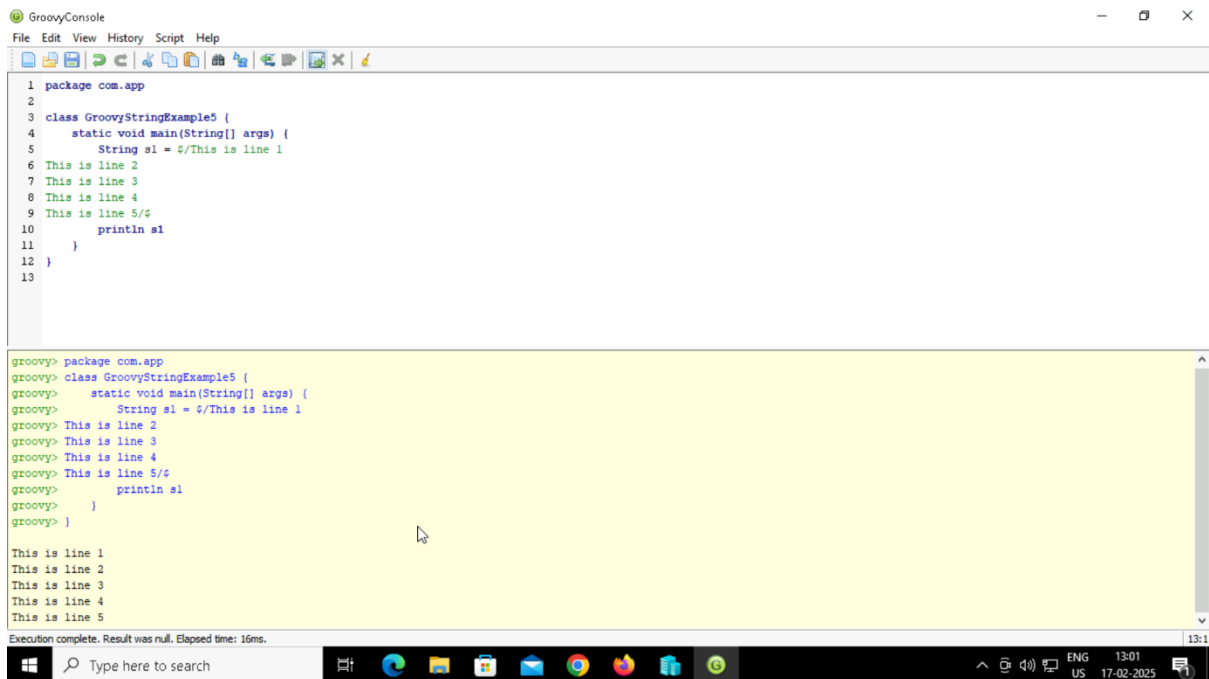
```
1 package com.app
2
3 class GroovyStringExample5 {
4     static void main(String[] args) {
5         String s1 = /This is a Groovy tutorial and we are learning strings/
6         println s1
7     }
8 }
9
```

```
groovy> package com.app
groovy> class GroovyStringExample5 {
groovy>     static void main(String[] args) {
groovy>         String s1 = /This is a Groovy tutorial and we are learning strings/
groovy>         println s1
groovy>     }
groovy> }

This is a Groovy tutorial and we are learning strings
```

Execution complete. Result was null. Elapsed time: 14ms.

Example 13



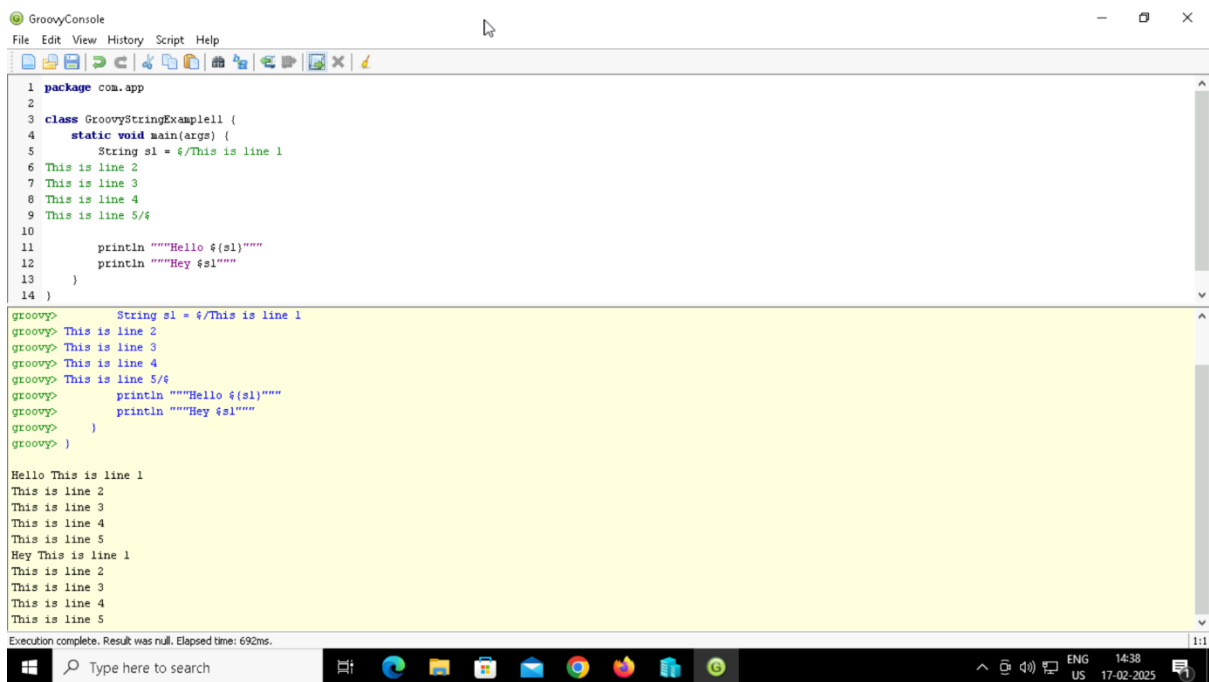
```
1 package com.app
2
3 class GroovyStringExample5 {
4     static void main(String[] args) {
5         String s1 = "This is line 1"
6         This is line 2
7         This is line 3
8         This is line 4
9         This is line 5
10        println s1
11    }
12 }
13
```

```
groovy> package com.app
groovy> class GroovyStringExample5 {
groovy>     static void main(String[] 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: 16ms.
```

Example 14



```
1 package com.app
2
3 class GroovyStringExample11 {
4     static void main(args) {
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        println ""Hello ${s1}""
12        println ""Hey ${s1}""
13    }
14 }

```

```
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 ""Hello ${s1}""
groovy> println ""Hey ${s1}""
groovy> }
groovy>

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: 692ms.
```

Groovy Project

5. Web Scraping with Groovy & Jsoup

Description:

Build a Groovy script that scrapes data from a website and saves it in a CSV file.

Solution:

- Use Jsoup to fetch and parse HTML content.
- Extract required elements using CSS selectors.
- Write the extracted data to a CSV file.

Tech Stack: Groovy, Jsoup, CSV Parsing

Use Case: Data Scraping, Automation

```
1 @Grab('org.jsoup:jsoup:1.16.2')
2
3 import org.jsoup.Jsoup
4 import java.nio.file.*
5
6 def url = "https://example.com"
7 def doc = Jsoup.connect(url).get()
8
9
10 def titles = doc.select("h2")
11 def csvContent = new StringBuilder("UST is a global digital technology company that provides IT and engineering services and solutions. They partner with clients to help them tr
12 titles.each { title ->
13     csvContent.append("${title.text()}\n")
14 }
15
16 def filePath = Paths.get("scraped_data.csv")
17 Files.write(filePath, csvContent.toString().getBytes())
18
19 println "Scraping complete! Data saved to scraped_data.csv"
20
```

Scraping complete! Data saved to scraped_data.csv

```
C:\Users\Administrator\Desktop>groovy -version
Groovy Version: 5.0.0-alpha-11 JVM: 11.0.15.1 Vendor: Oracle Corporation OS: Windows 10

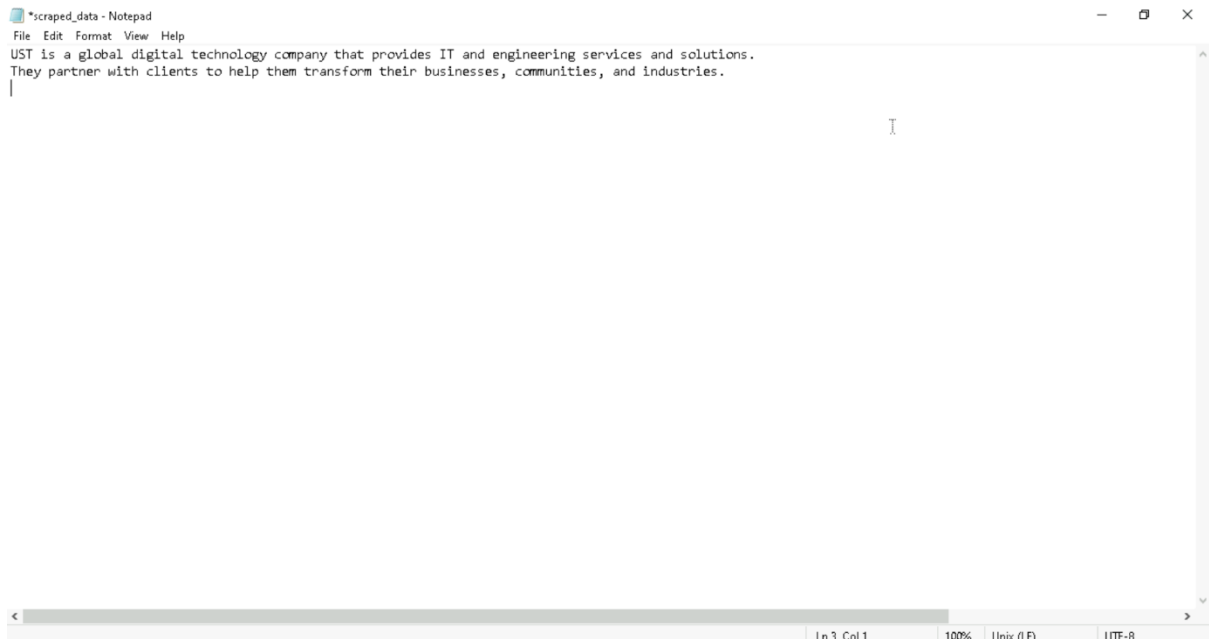
C:\Users\Administrator\Desktop>groovyConsole
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.codehaus.groovy.vmplugin.v9.Java9 (file:/C:/Program%20Files%20(x86)/Groovy/lib/groovy-5.0.0-alpha-11.jar) to method
java.awt.dnd.DropTargetContext.isDataFlavorSupported(java.awt.datatransfer.DataFlavor)
WARNING: Please consider reporting this to the maintainers of org.codehaus.groovy.vmplugin.v9.Java9
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
Terminate batch job (Y/N)? y

C:\Users\Administrator\Desktop>cd C:\Users\Administrator
C:\Users\Administrator>cd C:\Users\Administrator\Desktop
C:\Users\Administrator\Desktop>dir scraped_data.csv
Volume in drive C has no label.
Volume Serial Number is 5600-05A6

Directory of C:\Users\Administrator\Desktop
17-02-2025 17:43          197 scraped_data.csv
               1 File(s)          197 bytes
               0 Dir(s) 71,232,761,856 bytes free

C:\Users\Administrator\Desktop>notepad scraped_data.csv
C:\Users\Administrator\Desktop>
```

Output:



A screenshot of a Notepad window titled "*scraped_data - Notepad". The window contains the following text:

```
File Edit Format View Help
UST is a global digital technology company that provides IT and engineering services and solutions.
They partner with clients to help them transform their businesses, communities, and industries.
```

The status bar at the bottom indicates "Ln 3, Col 1", "100%", "Unix (LF)", and "UTF-8".

1. Groovy Script for Log File Analysis

Description:

Parse and analyze server log files to extract useful insights like error counts, request trends, and response times.

Solution:

- Read a log file line by line.
- Use regex to extract timestamp, status codes, and messages.
- Aggregate and summarize the data.
- Generate a report.

Tech Stack: Groovy, Regex, File I/O

Use Case: DevOps, System Monitoring



A screenshot of a terminal window showing the execution of a Groovy script for log file analysis. The terminal output is as follows:

```
pradhi@8b5c7dd85f01583: ~
pradhi@8b5c7dd85f01583:~$ nano server.log

GNU nano 7.2 server.log
2025-02-18 12:34:56 INFO 200 Request successful (100ms)
2025-02-18 12:35:00 ERROR 500 Internal Server Error (250ms)
2025-02-18 12:35:05 WARN 404 Not Found (50ms)
2025-02-18 12:35:10 INFO 200 Request successful (120ms)

pradhi@8b5c7dd85f01583:~$ nano log_parser.groovy
```

```
pradhi@8b5c7dd85f01583: ~
GNU nano 7.2 log_parser.groovy
import java.nio.file.Files
import java.nio.file.Paths
import java.util.regex.*

def logFilePath = "server.log" // Change this to your actual log file path

// Regex pattern to extract timestamp, status code, and message
def logPattern = ~/(\d{4}-\d{2}-\d{2} \d{2}:\d{2}:\d{2}) .* (\d{3}) (.+)/

def errorCount = [:].withDefault { 0 }
def statusCount = [:].withDefault { 0 }
def responseTimes = []

println "Processing log file: $logFilePath"

// Read file line by line
Files.lines(Paths.get(logFilePath)).each { line ->
    def matcher = logPattern.matcher(line)
    if (matcher.find()) {
        def timestamp = matcher.group(1)
        def statusCode = matcher.group(2)
        def message = matcher.group(3)

        // Count status codes
        statusCount[statusCode]++

        // Count errors (4xx and 5xx)
        if (statusCode.startsWith("4") || statusCode.startsWith("5")) {
            errorCount[statusCode]++
        }

        // Simulate extracting response time (if available in the log message)
        def responseTimeMatcher = message =~ /(\d+)ms/
        if (responseTimeMatcher.find()) {
            responseTimes << responseTimeMatcher.group(1).toInteger()
        }
    }
}

NG Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   ^U Undo       ^A Set Mark   ^] To Bracket
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^G Go To Line ^E Redo       ^G Copy       ^_ Where Was
```

Output:

```
pradhi@8b5c7dd85f01583: ~$ groovy log_parser.groovy
Processing log file: server.log

--- Log Analysis Report ---
Total Requests: 4
Status Code Counts: [200:2, 500:1, 404:1]
Error Counts: [500:1, 404:1]
Average Response Time: 130 ms
pradhi@8b5c7dd85f01583: ~$
```