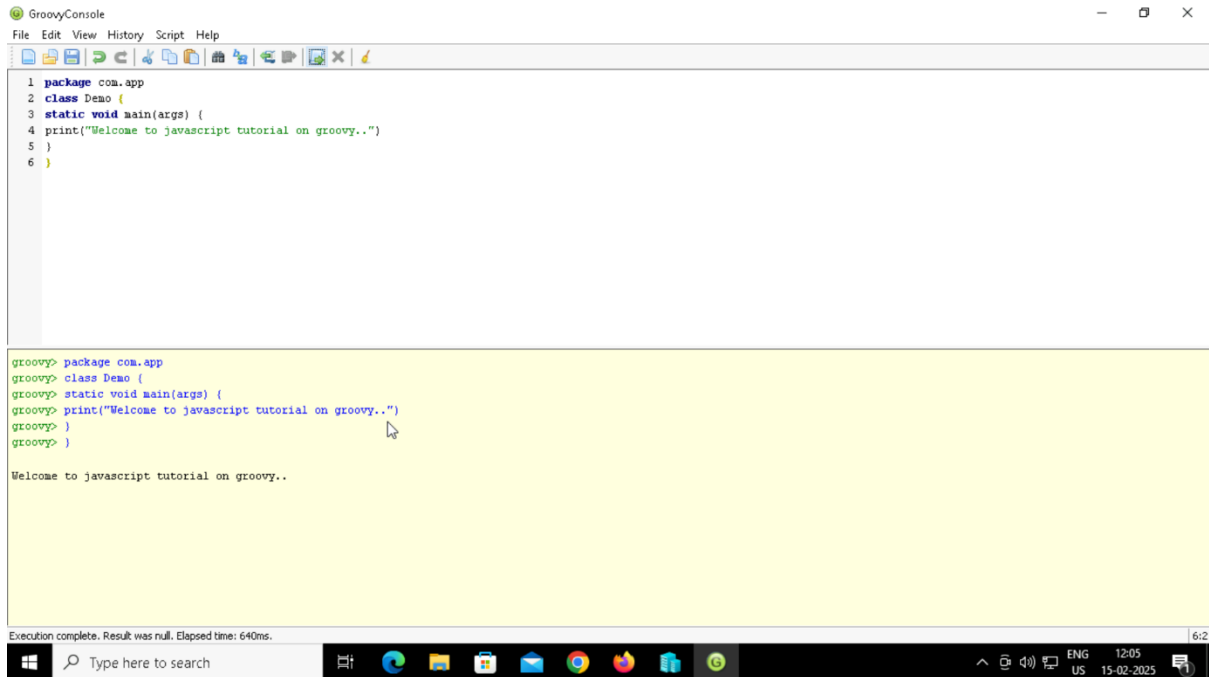


Groovy Assignment

First Groovy program



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 editor area contains the following code:

```
1 package com.app
2 class Demo {
3     static void main(args) {
4         print("Welcome to javascript tutorial on groovy..")
5     }
6 }
```

The console output area shows the command history and the result of the execution:

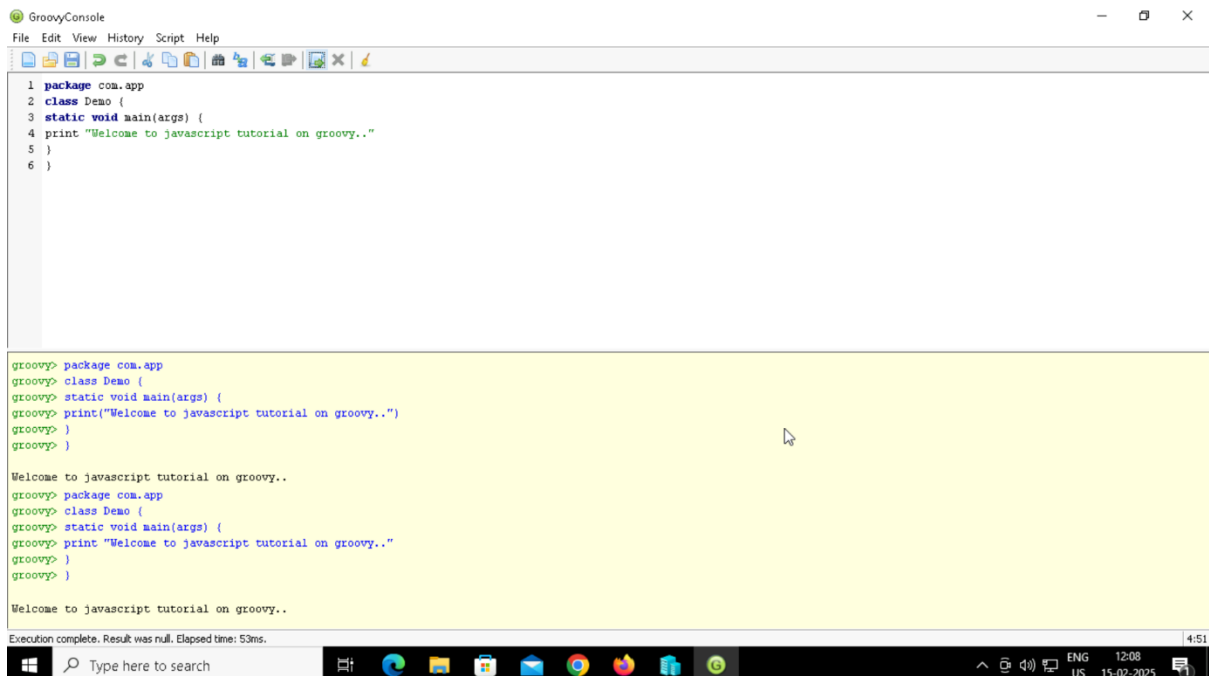
```
groovy> package com.app
groovy> class Demo {
groovy> static void main(args) {
groovy> print("Welcome to javascript tutorial on groovy..")
groovy> }
groovy> }

Welcome to javascript tutorial on groovy..
```

At the bottom, a status bar indicates "Execution complete. Result was null. Elapsed time: 640ms." The Windows taskbar is visible at the bottom of the screen.

Basic Syntax in groovy

Example 1



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 editor area contains the following code:

```
1 package com.app
2 class Demo {
3     static void main(args) {
4         print "Welcome to javascript tutorial on groovy.."
5     }
6 }
```

The console output area shows the command history and the result of the execution:

```
groovy> package com.app
groovy> class Demo {
groovy> static void main(args) {
groovy> print("Welcome to javascript tutorial on groovy..")
groovy> }
groovy> }

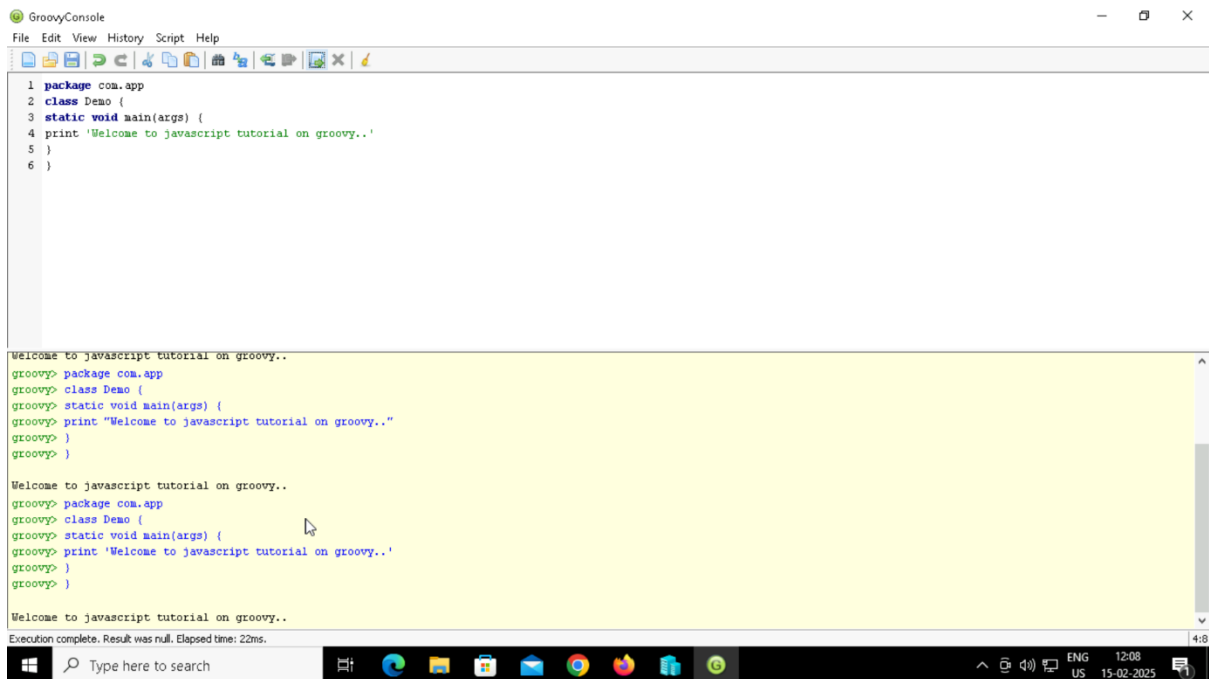
Welcome to javascript tutorial on groovy..

groovy> package com.app
groovy> class Demo {
groovy> static void main(args) {
groovy> print "Welcome to javascript tutorial on groovy.."
groovy> }
groovy> }

Welcome to javascript tutorial on groovy..
```

At the bottom, a status bar indicates "Execution complete. Result was null. Elapsed time: 53ms." The Windows taskbar is visible at the bottom of the screen.

Example 2



The screenshot shows the GroovyConsole application window. The title bar reads "GroovyConsole" with standard window controls. The menu bar includes "File", "Edit", "View", "History", "Script", and "Help". The toolbar contains icons for file operations and execution. The script editor contains the following code:

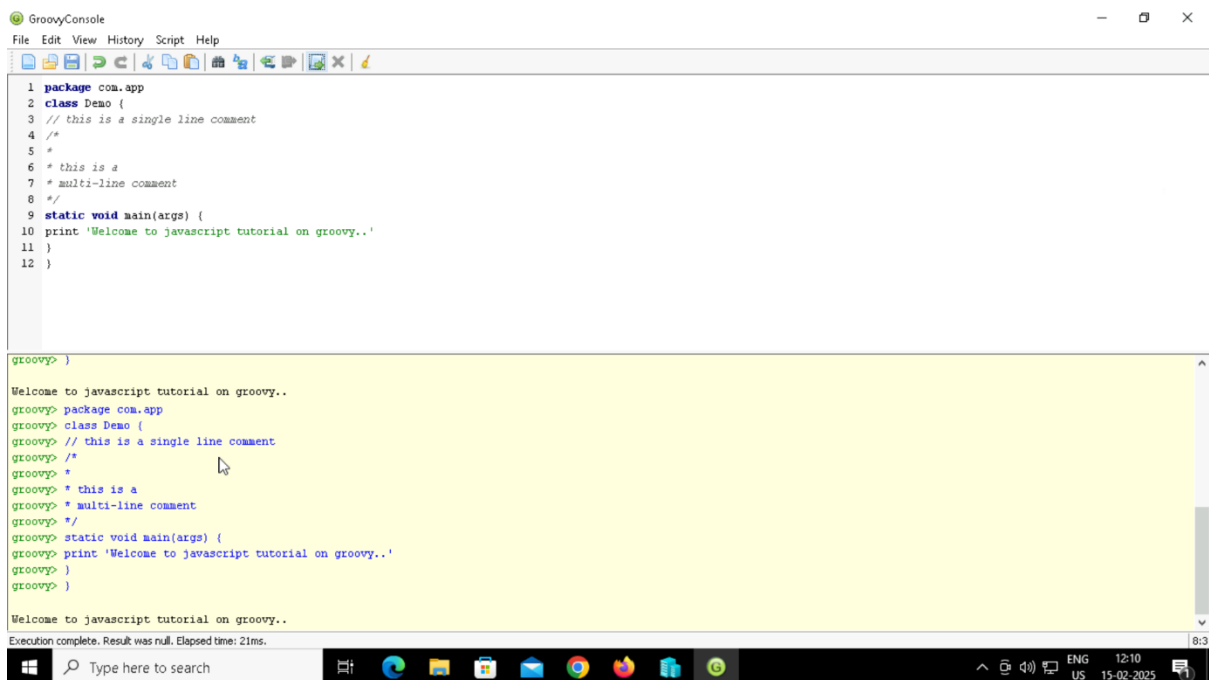
```
1 package com.app
2 class Demo {
3     static void main(args) {
4         print 'Welcome to javascript tutorial on groovy..'
5     }
6 }
```

The output pane shows the execution results:

```
Welcome to javascript tutorial on groovy..
groovy> package com.app
groovy> class Demo {
groovy> static void main(args) {
groovy> print "Welcome to javascript tutorial on groovy.."
groovy> }
groovy> }
```

Below the output, the status bar indicates: "Execution complete. Result was null. Elapsed time: 22ms." The Windows taskbar at the bottom shows the search bar and various application icons.

Example 3



The screenshot shows the GroovyConsole application window. The title bar reads "GroovyConsole" with standard window controls. The menu bar includes "File", "Edit", "View", "History", "Script", and "Help". The toolbar contains icons for file operations and execution. The script editor contains the following code:

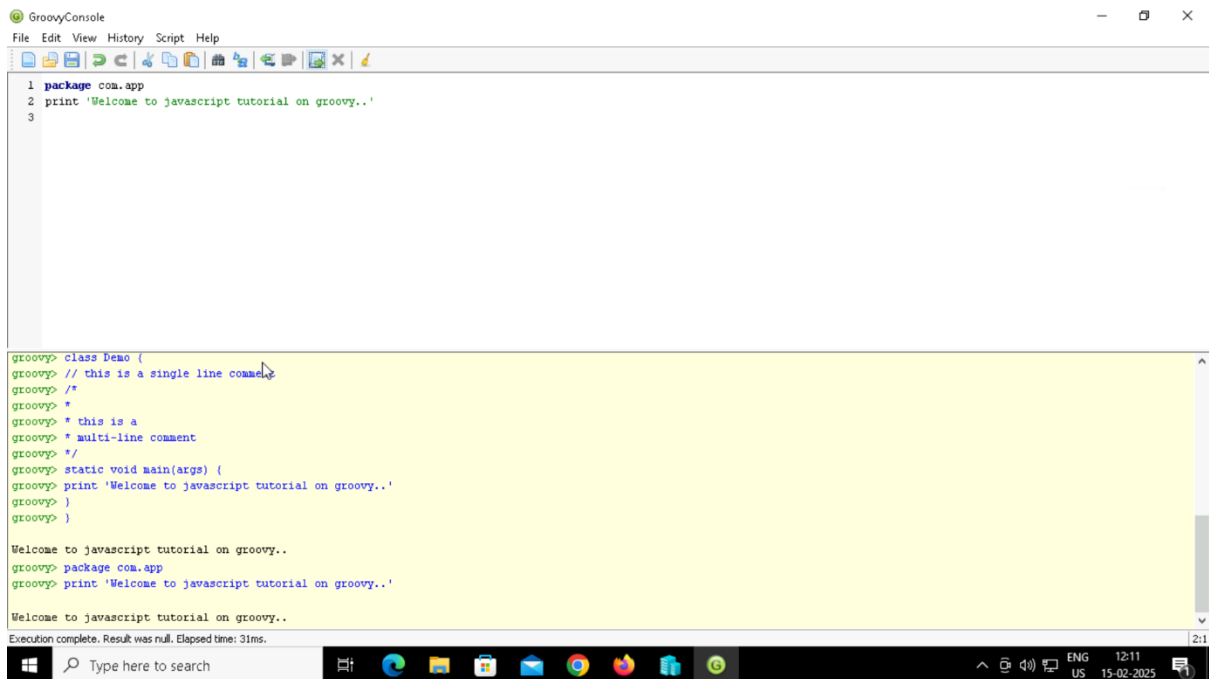
```
1 package com.app
2 class Demo {
3     // this is a single line comment
4     /*
5     *
6     * this is a
7     * multi-line comment
8     */
9     static void main(args) {
10        print 'Welcome to javascript tutorial on groovy..'
11    }
12 }
```

The output pane shows the execution results:

```
groovy> }
Welcome to javascript tutorial on groovy..
groovy> package com.app
groovy> class Demo {
groovy> // this is a single line comment
groovy> /*
groovy> *
groovy> * this is a
groovy> * multi-line comment
groovy> */
groovy> static void main(args) {
groovy> print 'Welcome to javascript tutorial on groovy..'
groovy> }
groovy> }
```

Below the output, the status bar indicates: "Execution complete. Result was null. Elapsed time: 21ms." The Windows taskbar at the bottom shows the search bar and various application icons.

Example 4



The screenshot shows the GroovyConsole application window. The title bar is 'GroovyConsole'. The menu bar includes 'File', 'Edit', 'View', 'History', 'Script', and 'Help'. The toolbar contains icons for file operations and execution. The script editor contains the following code:

```
1 package com.app
2 print 'Welcome to javascript tutorial on groovy..'
3
```

The output pane shows the execution results:

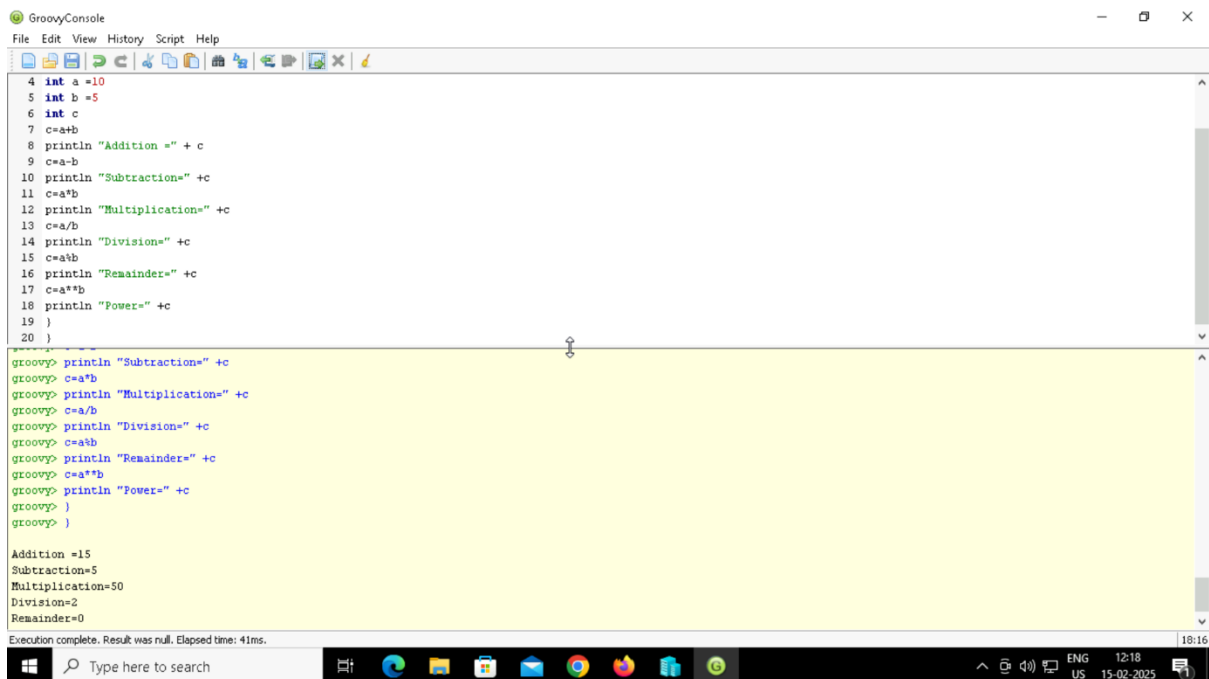
```
groovy> class Demo {
groovy> // this is a single line comment
groovy> /*
groovy> * this is a
groovy> * multi-line comment
groovy> */
groovy> static void main(args) {
groovy> print 'Welcome to javascript tutorial on groovy..'
groovy> }
groovy> }
```

Welcome to javascript tutorial on groovy..
groovy> package com.app
groovy> print 'Welcome to javascript tutorial on groovy..'

Welcome to javascript tutorial on groovy..
Execution complete. Result was null. Elapsed time: 31ms.

Arithmetic Operators

Example 1



The screenshot shows the GroovyConsole application window. The title bar is 'GroovyConsole'. The menu bar includes 'File', 'Edit', 'View', 'History', 'Script', and 'Help'. The toolbar contains icons for file operations and execution. The script editor contains the following code:

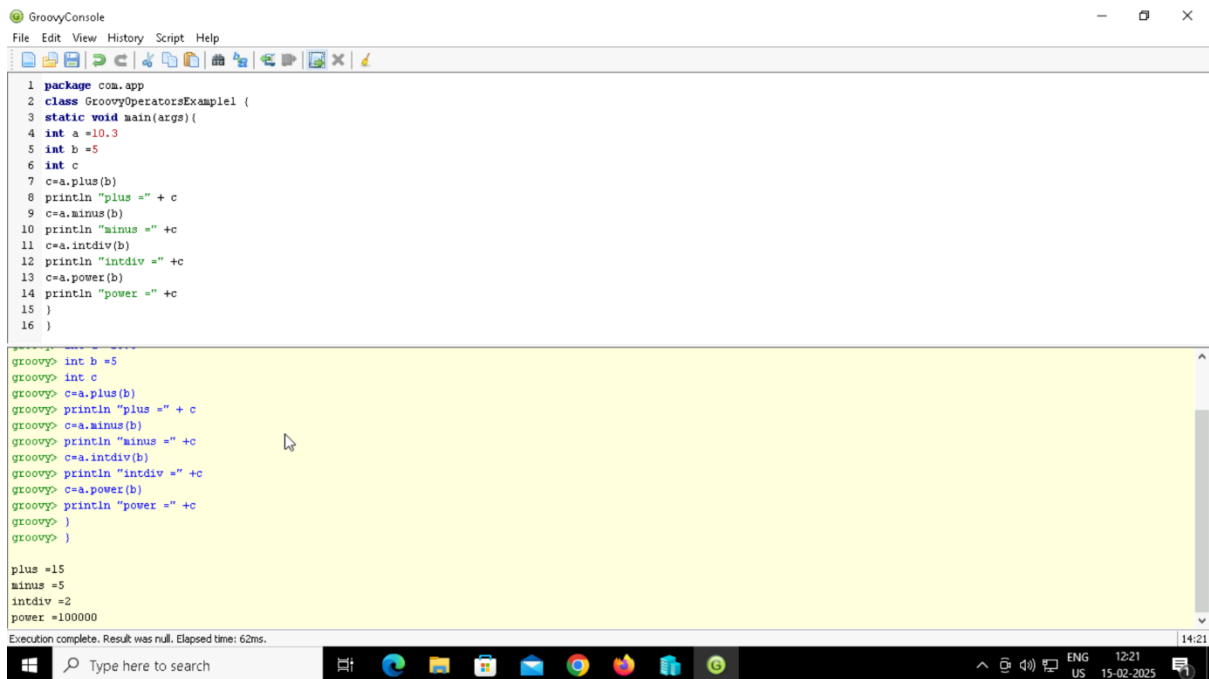
```
4 int a =10
5 int b =5
6 int c
7 c=a+b
8 println "Addition =" + c
9 c=a-b
10 println "Subtraction=" +c
11 c=a*b
12 println "Multiplication=" +c
13 c=a/b
14 println "Division=" +c
15 c=a%b
16 println "Remainder=" +c
17 c=a**b
18 println "Power=" +c
19 )
20 )
```

The output pane shows the execution results:

```
groovy> println "Subtraction=" +c
groovy> c=a*b
groovy> println "Multiplication=" +c
groovy> c=a/b
groovy> println "Division=" +c
groovy> c=a%b
groovy> println "Remainder=" +c
groovy> c=a**b
groovy> println "Power=" +c
groovy> )
groovy> )
```

Addition =15
Subtraction=5
Multiplication=50
Division=2
Remainder=0
Execution complete. Result was null. Elapsed time: 41ms.

Example 2



The screenshot shows the GroovyConsole application window. The title bar is 'GroovyConsole'. The menu bar includes 'File', 'Edit', 'View', 'History', 'Script', and 'Help'. The toolbar contains icons for file operations and execution. The script editor contains the following code:

```
1 package com.app
2 class GroovyOperatorsExample1 {
3     static void main(args) {
4         int a = 10.3
5         int b = 5
6         int c
7         c = a.plus(b)
8         println "plus =" + c
9         c = a.minus(b)
10        println "minus =" + c
11        c = a.intdiv(b)
12        println "intdiv =" + c
13        c = a.power(b)
14        println "power =" + c
15    }
16 }
```

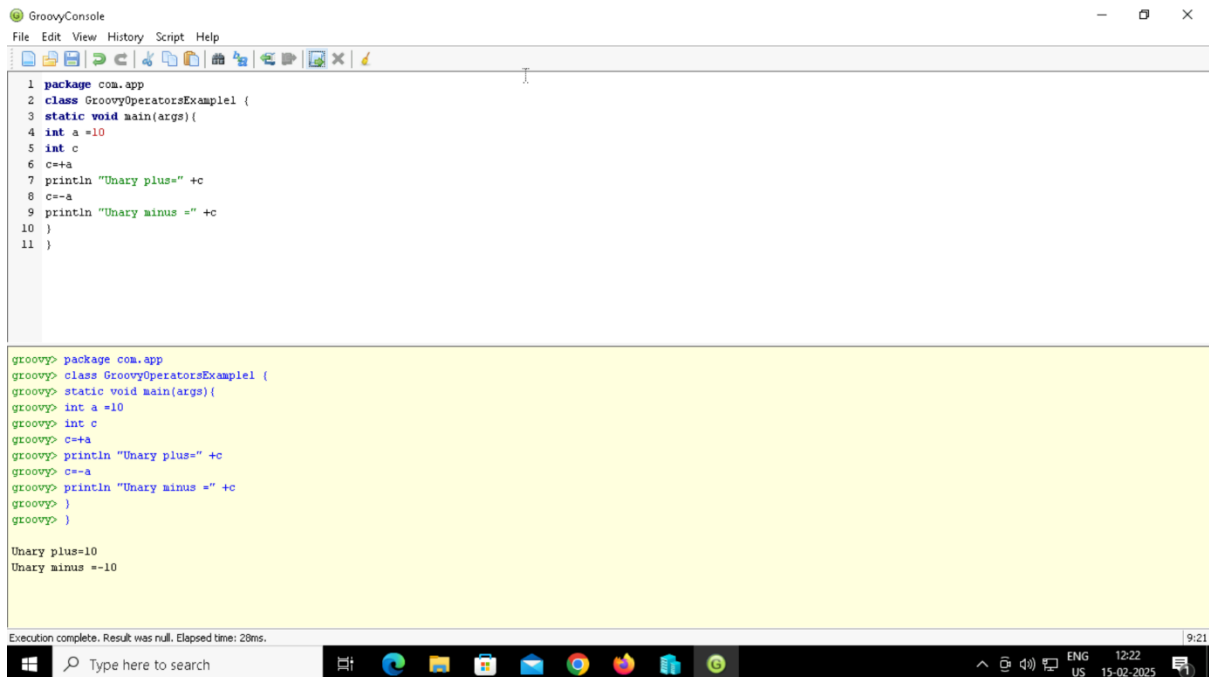
The output console shows the following results:

```
groovy> int b = 5
groovy> int c
groovy> c = a.plus(b)
groovy> println "plus =" + c
plus = 15
groovy> c = a.minus(b)
groovy> println "minus =" + c
minus = 5
groovy> c = a.intdiv(b)
groovy> println "intdiv =" + c
intdiv = 2
groovy> c = a.power(b)
groovy> println "power =" + c
power = 100000
groovy> }
```

Execution complete. Result was null. Elapsed time: 62ms.

Unary Operators

Example 3



The screenshot shows the GroovyConsole application window. The title bar is 'GroovyConsole'. The menu bar includes 'File', 'Edit', 'View', 'History', 'Script', and 'Help'. The toolbar contains icons for file operations and execution. The script editor contains the following code:

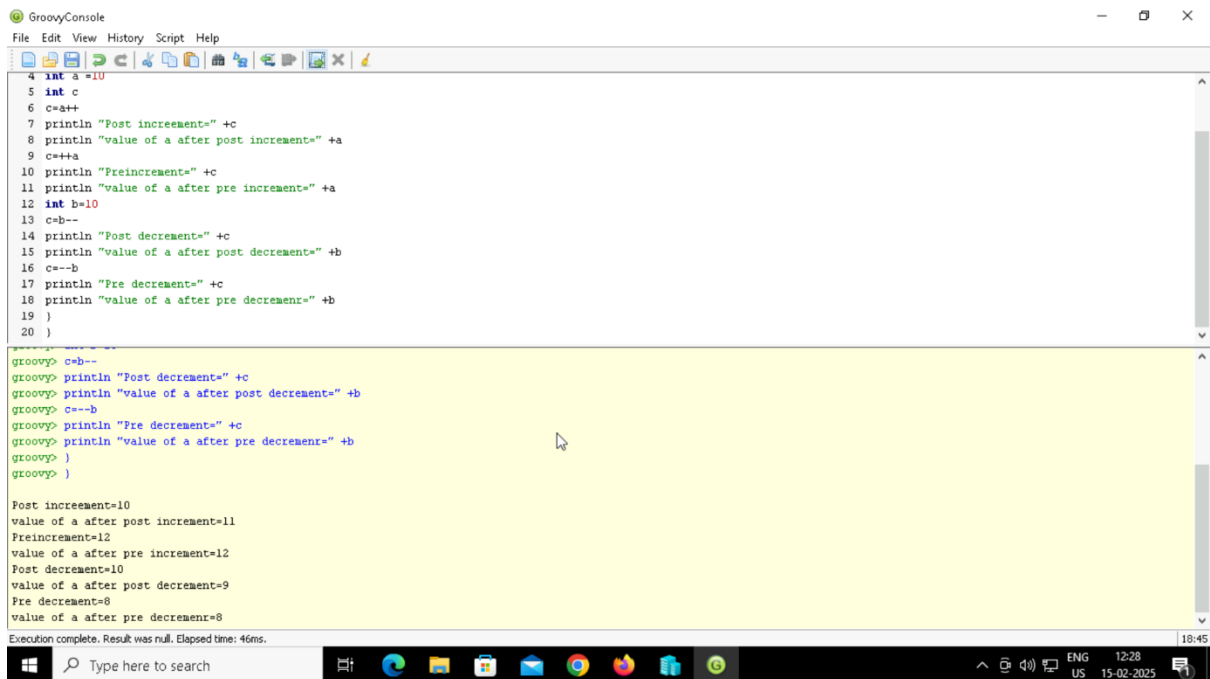
```
1 package com.app
2 class GroovyOperatorsExample1 {
3     static void main(args) {
4         int a = 10
5         int c
6         c = +a
7         println "Unary plus=" + c
8         c = -a
9         println "Unary minus =" + c
10    }
11 }
```

The output console shows the following results:

```
groovy> package com.app
groovy> class GroovyOperatorsExample1 {
groovy> static void main(args){
groovy> int a = 10
groovy> int c
groovy> c = +a
groovy> println "Unary plus=" + c
Unary plus=10
groovy> c = -a
groovy> println "Unary minus =" + c
Unary minus =-10
groovy> }
```

Execution complete. Result was null. Elapsed time: 28ms.

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 script:

```
4 int a = 10
5 int c
6 c = a++
7 println "Post increment=" + c
8 println "value of a after post increment=" + a
9 c += a
10 println "Preincrement=" + c
11 println "value of a after pre increment=" + a
12 int b = 10
13 c = b--
14 println "Post decrement=" + c
15 println "value of a after post decrement=" + b
16 c -= b
17 println "Pre decrement=" + c
18 println "value of a after pre decrement=" + b
19 }
20 }
```

The console output shows the results of the script execution:

```
groovy> c=b--
groovy> println "Post decrement=" + c
groovy> println "value of a after post decrement=" + b
groovy> c-=b
groovy> println "Pre decrement=" + c
groovy> println "value of a after pre decrement=" + b
groovy> }
groovy> }

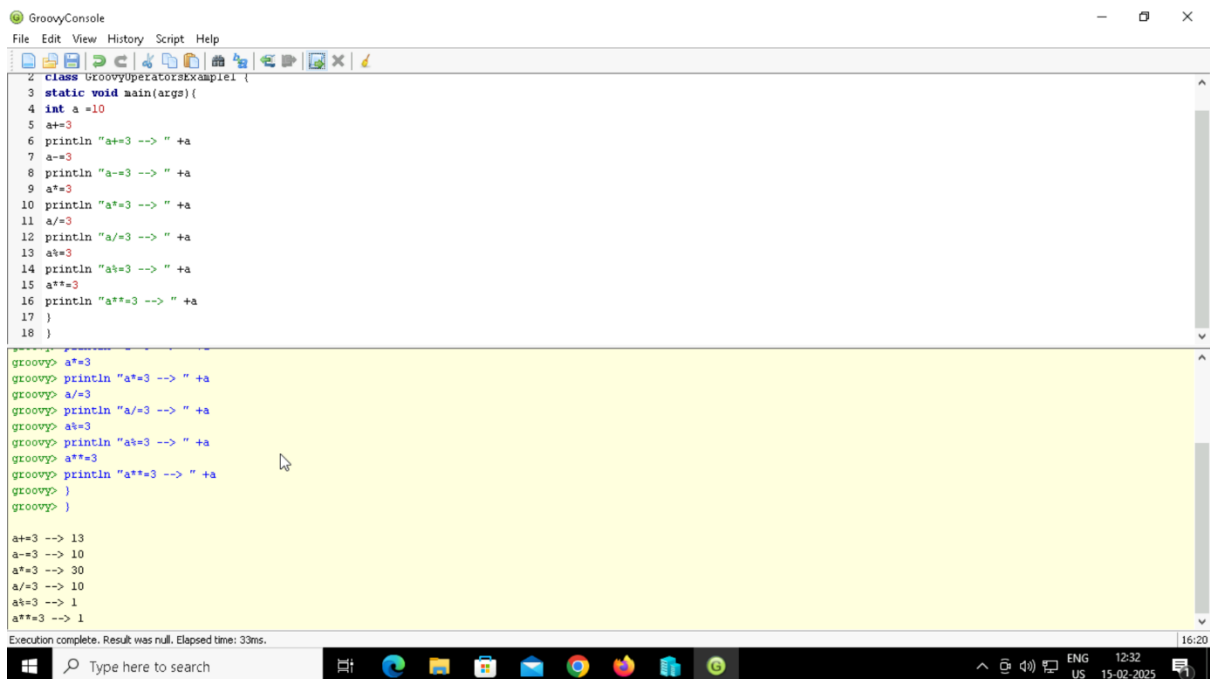
Post increment=10
value of a after post increment=11
Preincrement=12
value of a after pre increment=12
Post decrement=10
value of a after post decrement=9
Pre decrement=8
value of a after pre decrement=8

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

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

Assignment Arithmetic Operators

Example 5



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 script:

```
2 class GroovyOperatorsExample1 {
3     static void main(args) {
4         int a = 10
5         a += 3
6         println "a+=3 --> " + a
7         a -= 3
8         println "a-=3 --> " + a
9         a *= 3
10        println "a*=3 --> " + a
11        a /= 3
12        println "a/=3 --> " + a
13        a %= 3
14        println "a%=3 --> " + a
15        a **= 3
16        println "a**=3 --> " + a
17    }
18 }
```

The console output shows the results of the script execution:

```
groovy> a*=3
groovy> println "a*=3 --> " + a
groovy> a/=3
groovy> println "a/=3 --> " + a
groovy> a%=3
groovy> println "a%=3 --> " + a
groovy> a**=3
groovy> println "a**=3 --> " + a
groovy> }
groovy> }

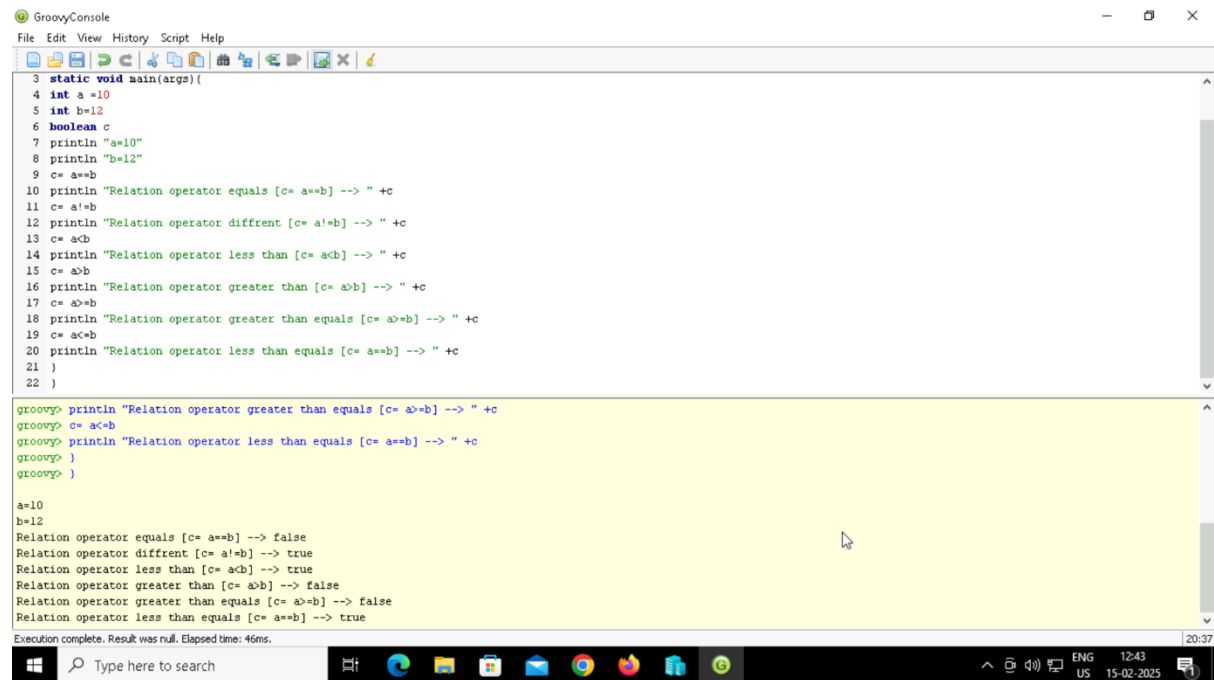
a+=3 --> 13
a-=3 --> 10
a*=3 --> 30
a/=3 --> 10
a%=3 --> 1
a**=3 --> 1

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

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

Relational Operators

Example 6



The screenshot shows the GroovyConsole application with a script for relational operators. The script defines variables `a=10` and `b=12`, and a boolean `c`. It then prints the results of various relational operations: `a==b` (false), `a!=b` (true), `a<b` (true), `a>b` (false), `a>=b` (false), and `a<=b` (true). The console output shows the results of these operations, confirming the expected behavior of the relational operators.

```
3 static void main(args){
4   int a =10
5   int b=12
6   boolean c
7   println "a=10"
8   println "b=12"
9   c= a==b
10  println "Relation operator equals [c= a==b] --> " +c
11  c= a!=b
12  println "Relation operator diffrent [c= a!=b] --> " +c
13  c= a<b
14  println "Relation operator less than [c= a<b] --> " +c
15  c= a>b
16  println "Relation operator greater than [c= a>b] --> " +c
17  c= a>=b
18  println "Relation operator greater than equals [c= a>=b] --> " +c
19  c= a<=b
20  println "Relation operator less than equals [c= a<=b] --> " +c
21 }
22 }
```

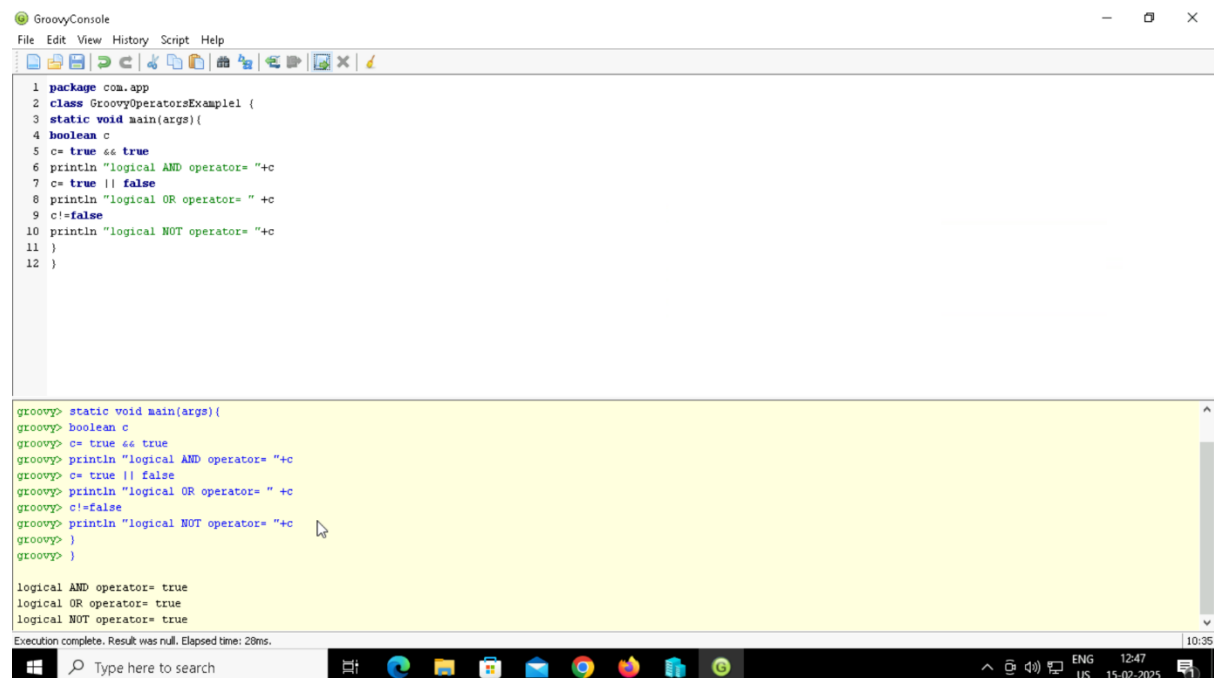
```
groovy> println "Relation operator greater than equals [c= a>=b] --> " +c
groovy> c= a<=b
groovy> println "Relation operator less than equals [c= a<=b] --> " +c
groovy> }
groovy> }

a=10
b=12
Relation operator equals [c= a==b] --> false
Relation operator diffrent [c= a!=b] --> true
Relation operator less than [c= a<b] --> true
Relation operator greater than [c= a>b] --> false
Relation operator greater than equals [c= a>=b] --> false
Relation operator less than equals [c= a<=b] --> true

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

Logical operators

Example 7



The screenshot shows the GroovyConsole application with a script for logical operators. The script defines a boolean `c` and then prints the results of logical operations: `c=true && true` (true), `c=true || false` (true), `c!=false` (true), and `c=true` (true). The console output shows the results of these operations, confirming the expected behavior of the logical operators.

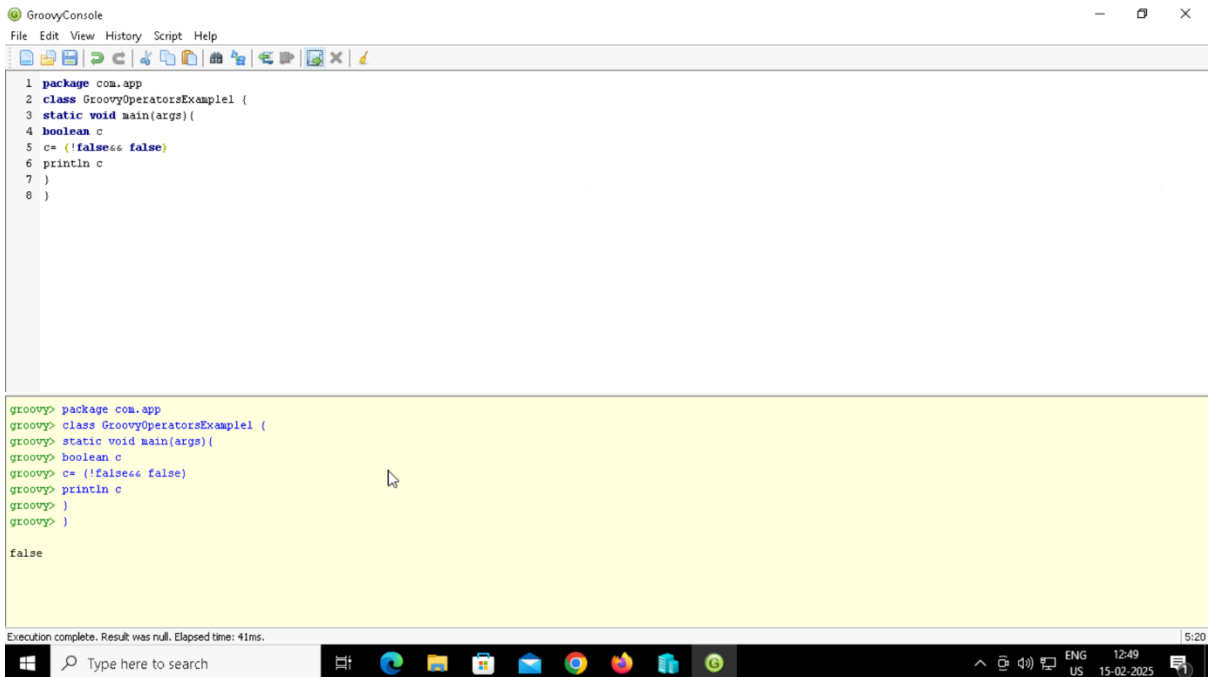
```
1 package com.app
2 class GroovyOperatorsExampel {
3   static void main(args){
4     boolean c
5     c= true && true
6     println "logical AND operator= "+c
7     c= true || false
8     println "logical OR operator= " +c
9     c!=false
10    println "logical NOT operator= "+c
11  }
12 }
```

```
groovy> static void main(args){
groovy> boolean c
groovy> c= true && true
groovy> println "logical AND operator= "+c
groovy> c= true || false
groovy> println "logical OR operator= " +c
groovy> c!=false
groovy> println "logical NOT operator= "+c
groovy> }
groovy> }

logical AND operator= true
logical OR operator= true
logical NOT operator= true

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

Example 8

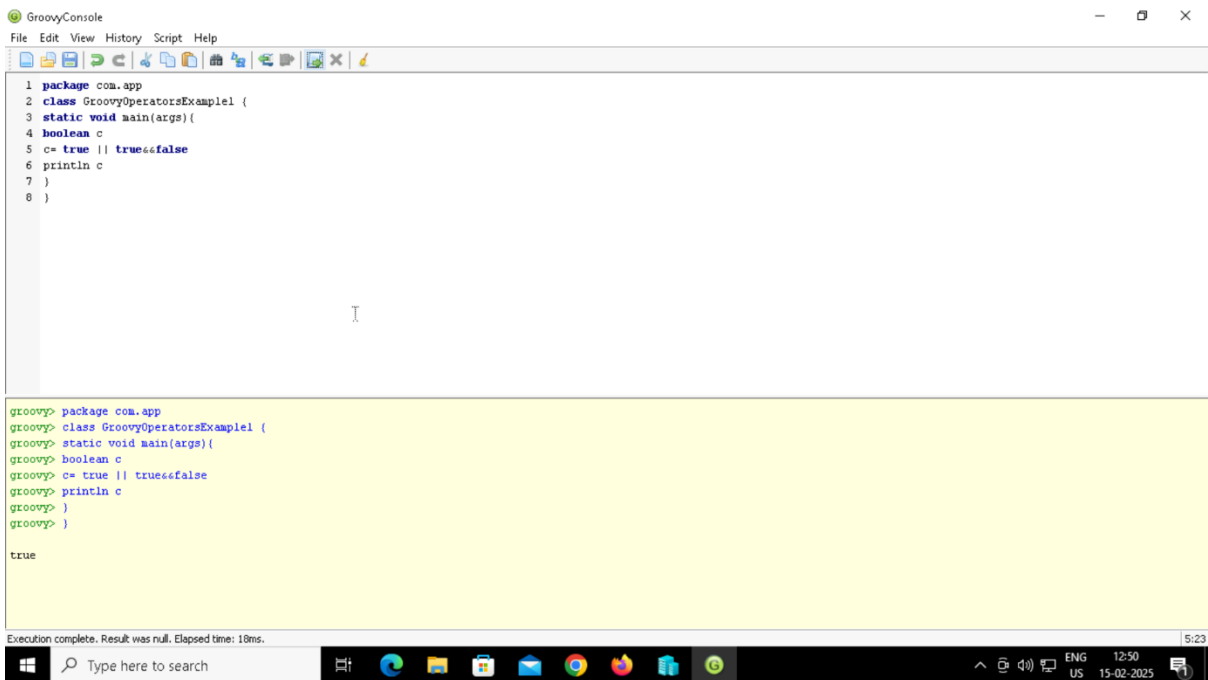


```
1 package com.app
2 class GroovyOperatorsExample1 {
3     static void main(args){
4         boolean c
5         c = (!false == false)
6         println c
7     }
8 }
```

```
groovy> package com.app
groovy> class GroovyOperatorsExample1 {
groovy> static void main(args){
groovy> boolean c
groovy> c = (!false == false)
groovy> println c
groovy> }
false
```

Execution complete. Result was null. Elapsed time: 41ms.

Example 9



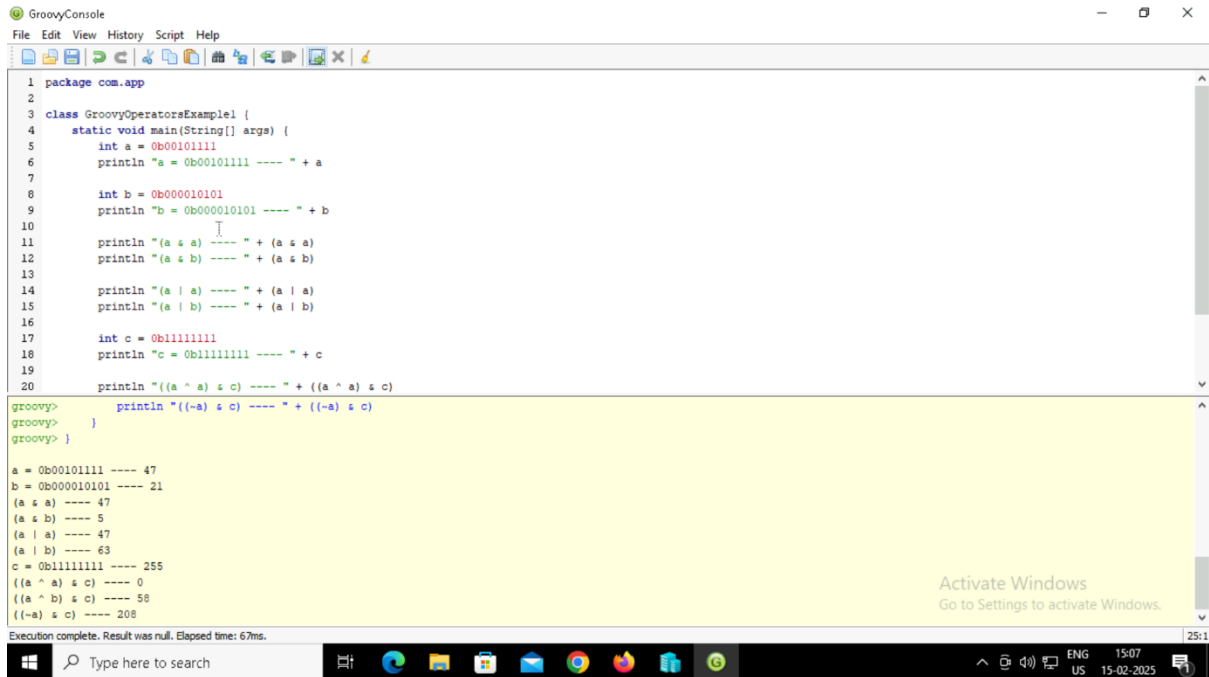
```
1 package com.app
2 class GroovyOperatorsExample1 {
3     static void main(args){
4         boolean c
5         c = true || true == false
6         println c
7     }
8 }
```

```
groovy> package com.app
groovy> class GroovyOperatorsExample1 {
groovy> static void main(args){
groovy> boolean c
groovy> c = true || true == false
groovy> println c
groovy> }
true
```

Execution complete. Result was null. Elapsed time: 18ms.

Bitwise operators

Example 10



The screenshot shows a GroovyConsole window with a menu bar (File, Edit, View, History, Script, Help) and a toolbar. The code in the editor is as follows:

```
1 package com.app
2
3 class GroovyOperatorsExample1 {
4     static void main(String[] args) {
5         int a = 0b00101111
6         println "a = 0b00101111 ---- " + a
7
8         int b = 0b000010101
9         println "b = 0b000010101 ---- " + b
10
11         println "(a & a) ---- " + (a & a)
12         println "(a & b) ---- " + (a & b)
13
14         println "(a | a) ---- " + (a | a)
15         println "(a | b) ---- " + (a | b)
16
17         int c = 0b11111111
18         println "c = 0b11111111 ---- " + c
19
20         println "((a ^ a) & c) ---- " + ((a ^ a) & c)
21
22         println "((-a) & c) ---- " + ((-a) & c)
23     }
24 }
```

The output in the console is:

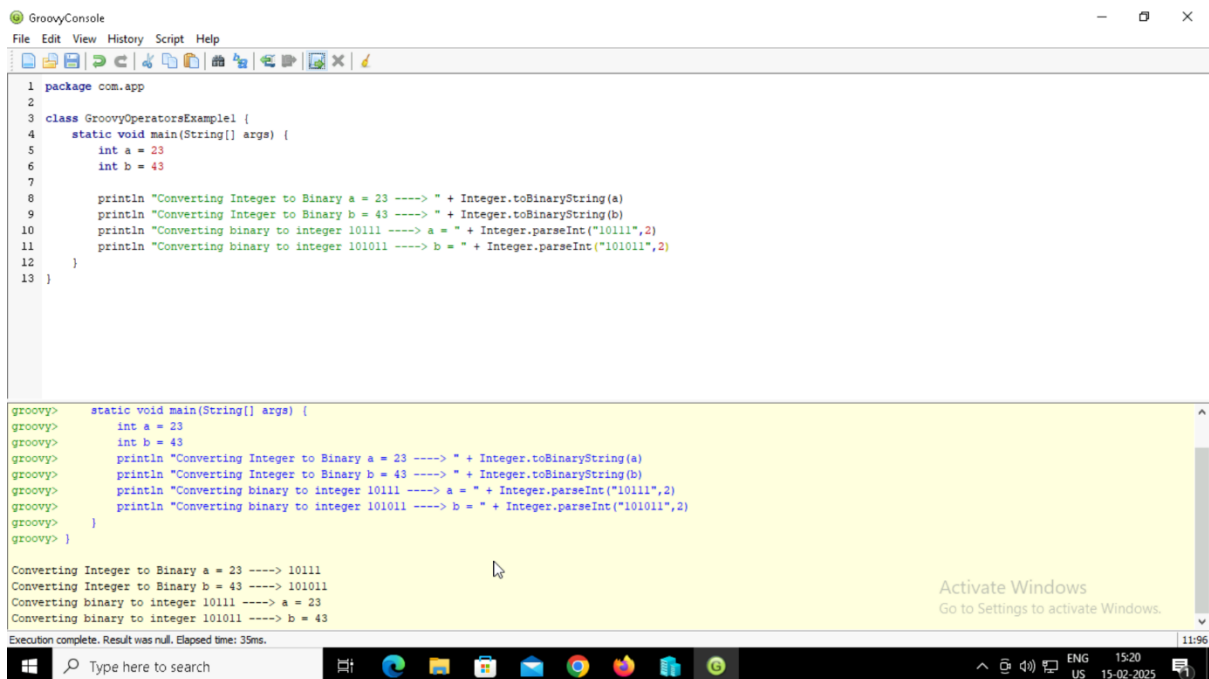
```
groovy>
groovy>
groovy> ]
groovy> ]

a = 0b00101111 ---- 47
b = 0b000010101 ---- 21
(a & a) ---- 47
(a & b) ---- 5
(a | a) ---- 47
(a | b) ---- 63
c = 0b11111111 ---- 255
((a ^ a) & c) ---- 0
((a ^ b) & c) ---- 58
((-a) & c) ---- 208

Execution complete. Result was null. Elapsed time: 67ms.
```

An "Activate Windows" watermark is visible in the bottom right corner of the console output area.

Example 11



The screenshot shows a GroovyConsole window with a menu bar (File, Edit, View, History, Script, Help) and a toolbar. The code in the editor is as follows:

```
1 package com.app
2
3 class GroovyOperatorsExample1 {
4     static void main(String[] args) {
5         int a = 23
6         int b = 43
7
8         println "Converting Integer to Binary a = 23 ----> " + Integer.toBinaryString(a)
9         println "Converting Integer to Binary b = 43 ----> " + Integer.toBinaryString(b)
10        println "Converting binary to integer 1011 ----> a = " + Integer.parseInt("1011",2)
11        println "Converting binary to integer 101011 ----> b = " + Integer.parseInt("101011",2)
12    }
13 }
```

The output in the console is:

```
groovy>
groovy>
groovy>
groovy>
groovy>
groovy>
groovy>
groovy>
groovy>
groovy>
groovy>
groovy>
groovy> ]
groovy> ]

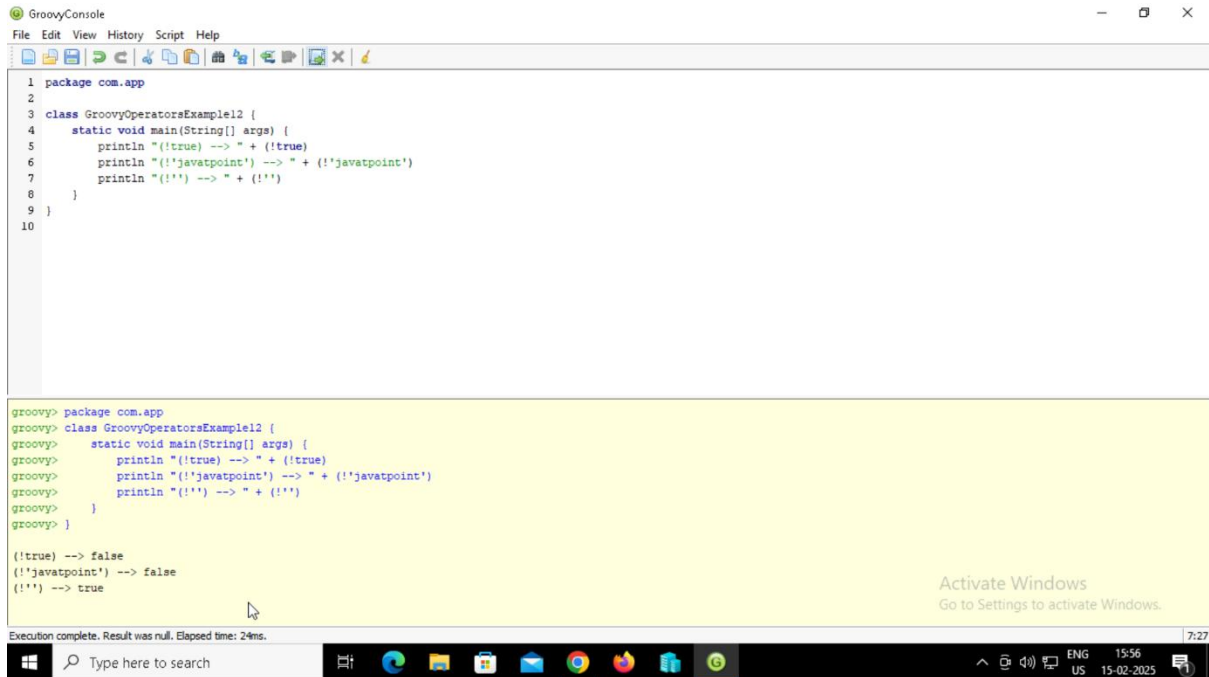
Converting Integer to Binary a = 23 ----> 1011
Converting Integer to Binary b = 43 ----> 101011
Converting binary to integer 1011 ----> a = 23
Converting binary to integer 101011 ----> b = 43

Execution complete. Result was null. Elapsed time: 35ms.
```

An "Activate Windows" watermark is visible in the bottom right corner of the console output area.

Conditional operators

Example 12



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
3 class GroovyOperatorsExample12 {
4     static void main(String[] args) {
5         println "(!true) --> " + (!true)
6         println "(!'javatpoint') --> " + (!'javatpoint')
7         println "(!'') --> " + (!'')
8     }
9 }
10
```

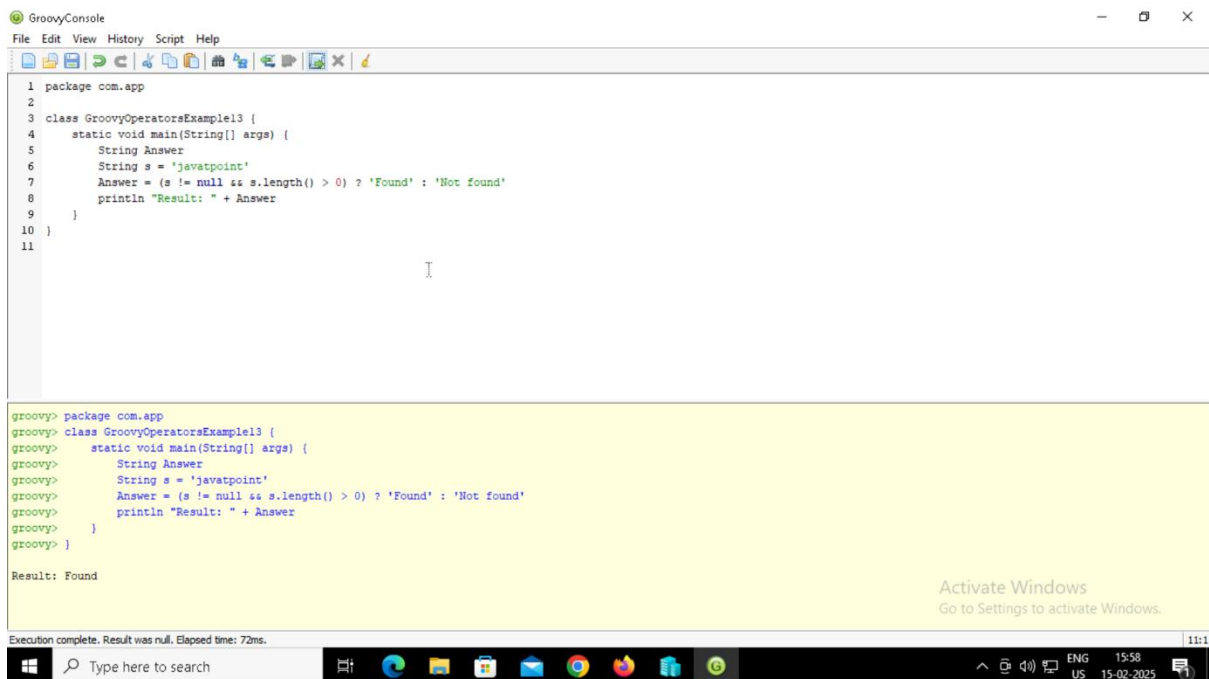
The console output shows the execution results:

```
groovy> package com.app
groovy> class GroovyOperatorsExample12 {
groovy>     static void main(String[] args) {
groovy>         println "(!true) --> " + (!true)
groovy>         println "(!'javatpoint') --> " + (!'javatpoint')
groovy>         println "(!'') --> " + (!'')
groovy>     }
groovy> }

(!true) --> false
(!'javatpoint') --> false
(!'') --> true
```

The status bar at the bottom indicates "Execution complete. Result was null. Elapsed time: 24ms." and the system tray shows the date and time as 15:56 on 15-02-2025.

Example 13



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
3 class GroovyOperatorsExample13 {
4     static void main(String[] args) {
5         String Answer
6         String s = 'javatpoint'
7         Answer = (s != null && s.length() > 0) ? 'Found' : 'Not found'
8         println "Result: " + Answer
9     }
10 }
11
```

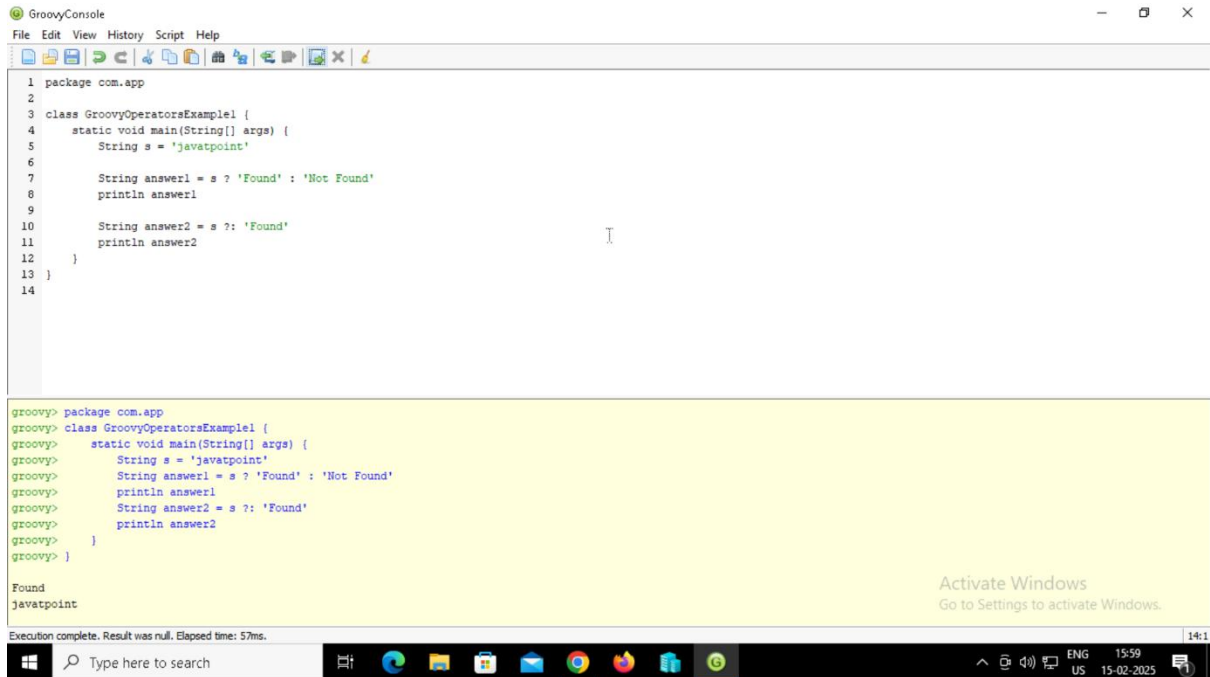
The console output shows the execution results:

```
groovy> package com.app
groovy> class GroovyOperatorsExample13 {
groovy>     static void main(String[] args) {
groovy>         String Answer
groovy>         String s = 'javatpoint'
groovy>         Answer = (s != null && s.length() > 0) ? 'Found' : 'Not found'
groovy>         println "Result: " + Answer
groovy>     }
groovy> }

Result: Found
```

The status bar at the bottom indicates "Execution complete. Result was null. Elapsed time: 72ms." and the system tray shows the date and time as 15:58 on 15-02-2025.

Example 14



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 script editor displays the following code:

```
1 package com.app
2
3 class GroovyOperatorsExample1 {
4     static void main(String[] args) {
5         String s = 'javatpoint'
6
7         String answer1 = s ? 'Found' : 'Not Found'
8         println answer1
9
10        String answer2 = s ? 'Found'
11        println answer2
12    }
13 }
14
```

The output pane shows the execution results:

```
groovy> package com.app
groovy> class GroovyOperatorsExample1 {
groovy>     static void main(String[] args) {
groovy>         String s = 'javatpoint'
groovy>         String answer1 = s ? 'Found' : 'Not Found'
groovy>         println answer1
groovy>         String answer2 = s ? 'Found'
groovy>         println answer2
groovy>     }
groovy> }
Found
javatpoint
```

At the bottom of the output pane, it says "Execution complete. Result was null. Elapsed time: 57ms." and "14:1".

On the right side of the output pane, there is a watermark that says "Activate Windows Go to Settings to activate Windows."

The Windows taskbar is visible at the bottom, showing the search bar and various application icons. The system clock shows "15:59 15-02-2025".