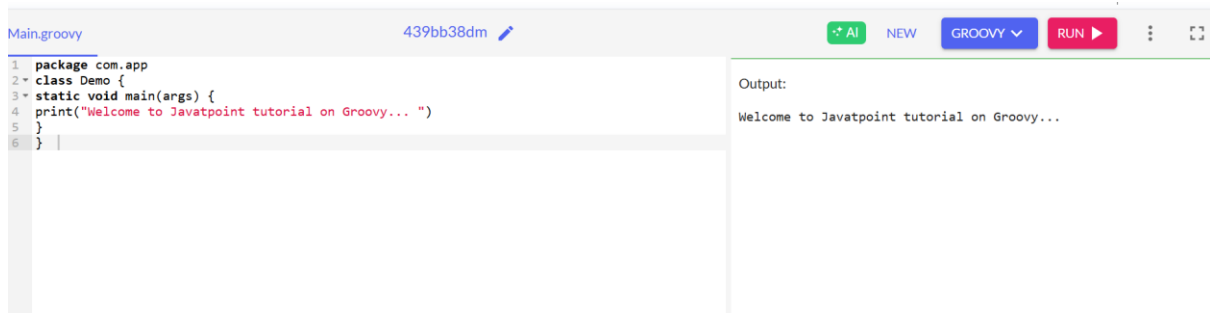


GROOVY PROGRAMS ASSIGNMENT

First Groovy Program



The screenshot shows an IDE window titled 'Main.groovy' with a file icon and a blue pencil icon. The code is as follows:

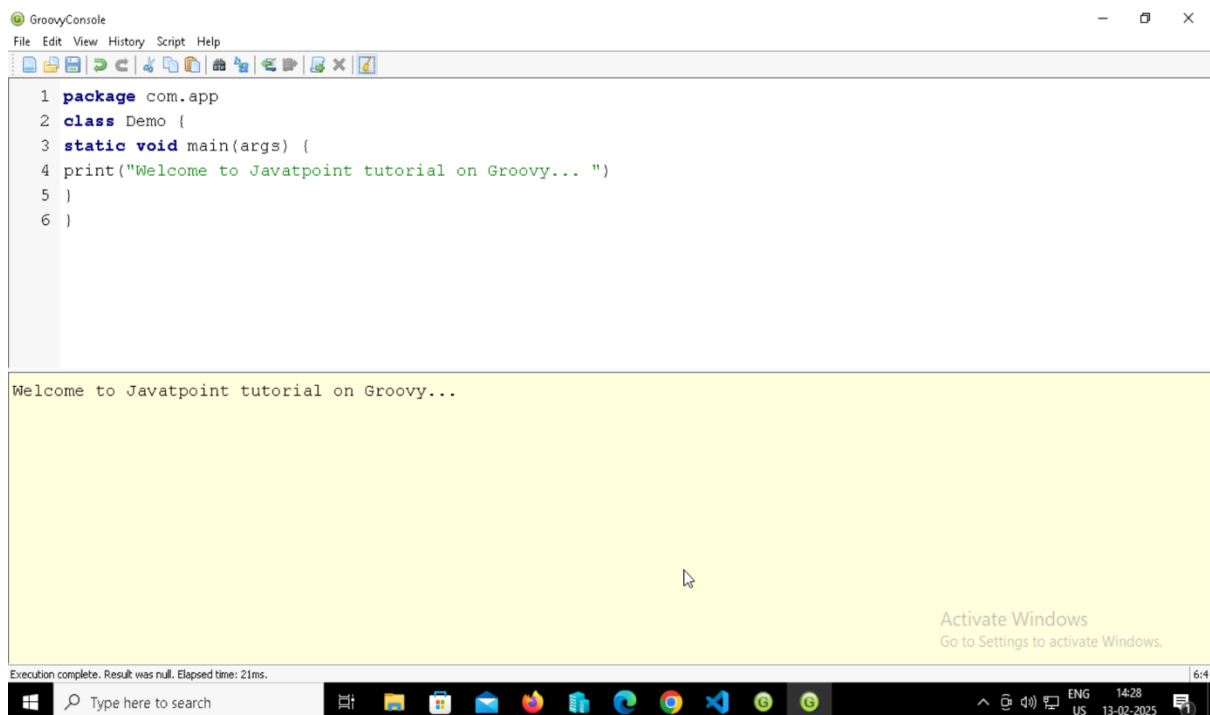
```
1 package com.app
2 class Demo {
3     static void main(args) {
4         print("Welcome to Javatpoint tutorial on Groovy... ")
5     }
6 }
```

On the right side, there is a 'GROOVY' dropdown menu and a 'RUN' button with a play icon. Below these, the output is displayed:

Output:
Welcome to Javatpoint tutorial on Groovy...

Groovy basic syntax

Example programs



The screenshot shows a 'GroovyConsole' application window. The code is as follows:

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

The output is displayed in a yellow box:

Welcome to Javatpoint tutorial on Groovy...

At the bottom, a status bar indicates: 'Execution complete. Result was null. Elapsed time: 21ms.'

The Windows taskbar is visible at the bottom, showing the search bar, taskbar icons, and system tray with the date '13-02-2025' and time '14:28'.

```
MainGroovy 439bb38dm
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 Javatpoint tutorial on Groovy..."
11 }
12 }
13
14
15
```

Output:

Welcome to Javatpoint tutorial on Groovy...

GROOVY OPERATORS

Groovy unary operator

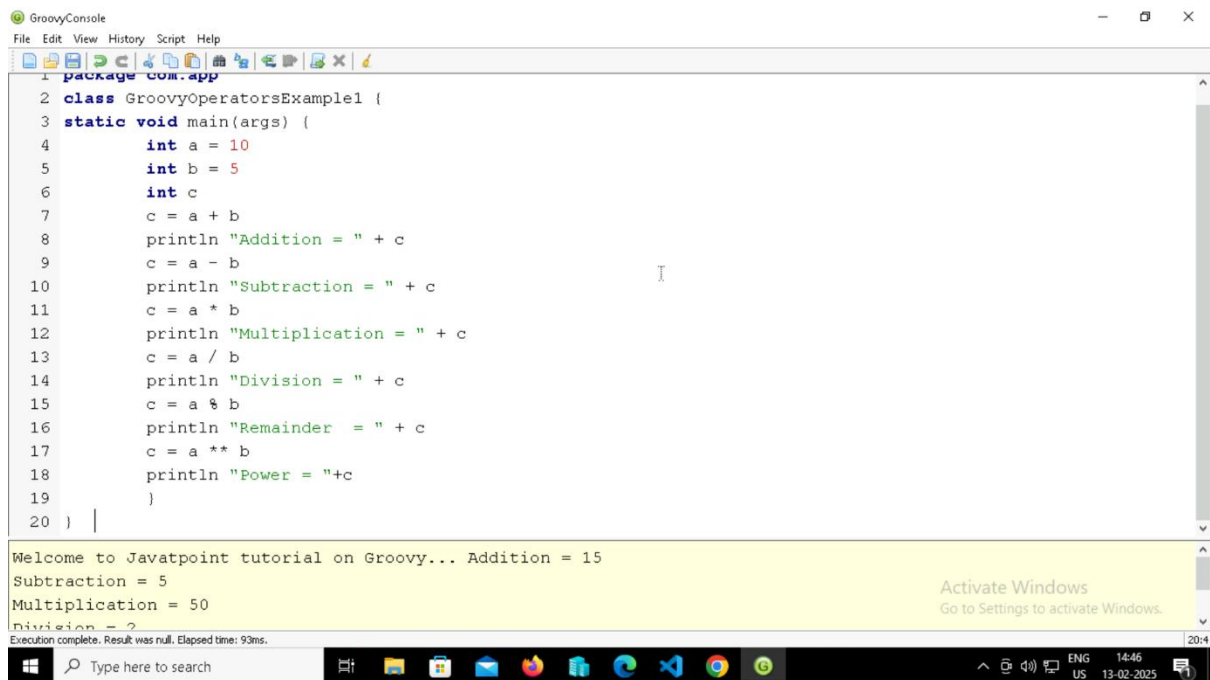
```
GroovyConsole
File Edit View History Script Help
1 package com.app
2 class GroovyOperatorsExample3 {
3 static void main(args) {
4     int a = 10
5     int c
6     c = +a
7     println "Unary plus = " + c
8     c = -a
9     println "Unary minus = " + c
10
11 }
12 }
```

Unary plus = 10
Unary minus = -10

Execution complete. Result was null. Elapsed time: 31ms.

Activate Windows
Go to Settings to activate Windows.

Groovy arithmetic operators

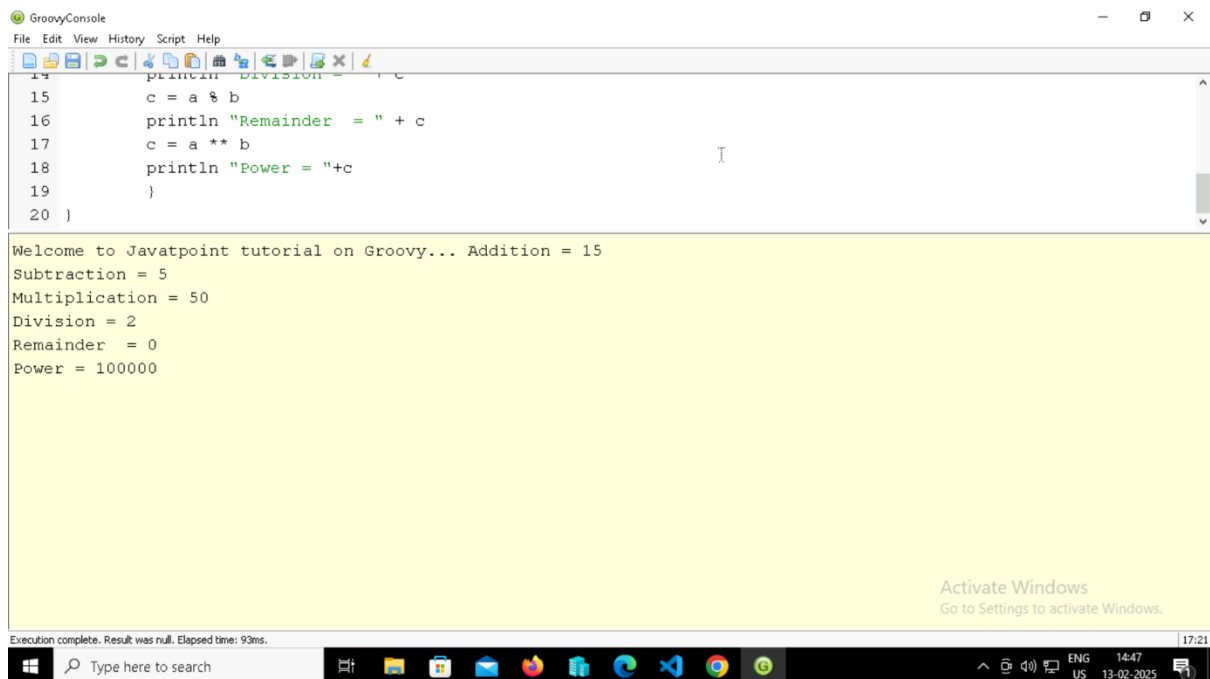


The screenshot shows the GroovyConsole application with a script for arithmetic operations. The script defines a class `GroovyOperatorsExample1` with a `main` method. It initializes `a = 10` and `b = 5`, then performs addition, subtraction, multiplication, division, and modulo operations, printing the results. The output shows the results of these operations: Addition = 15, Subtraction = 5, Multiplication = 50, Division = 2, and Remainder = 0. The execution is complete, and the elapsed time is 93ms.

```
1 package com.app
2 class GroovyOperatorsExample1 {
3     static void main(args) {
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 }
```

Welcome to Javatpoint tutorial on Groovy... Addition = 15
Subtraction = 5
Multiplication = 50
Division = 2
Remainder = 0
Power = 100000

Execution complete. Result was null. Elapsed time: 93ms.



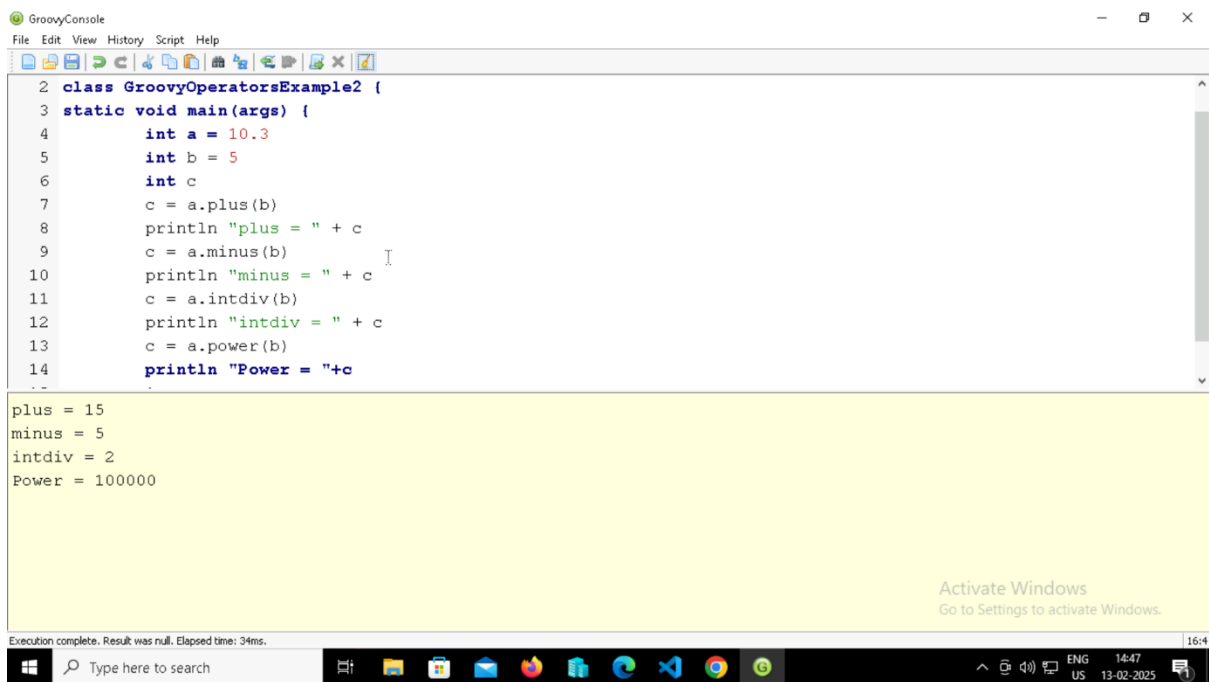
The screenshot shows the GroovyConsole application with a Groovy script for arithmetic operations. The script defines a class `GroovyOperatorsExample1` with a `main` method. It initializes `a = 10` and `b = 5`, then performs addition, subtraction, multiplication, division, and modulo operations, printing the results. The output shows the results of these operations: Addition = 15, Subtraction = 5, Multiplication = 50, Division = 2, Remainder = 0, and Power = 100000. The execution is complete, and the elapsed time is 93ms.

```
14 println "Division = " + c
15 c = a % b
16 println "Remainder = " + c
17 c = a ** b
18 println "Power = "+c
19 }
20 }
```

Welcome to Javatpoint tutorial on Groovy... Addition = 15
Subtraction = 5
Multiplication = 50
Division = 2
Remainder = 0
Power = 100000

Execution complete. Result was null. Elapsed time: 93ms.

Groovy arithmetic operator example 2



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:

```
2 class GroovyOperatorsExample2 {
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 }
```

Below the code, the output is displayed on a yellow background:

```
plus = 15
minus = 5
intdiv = 2
Power = 100000
```

At the bottom, a status bar indicates "Execution complete. Result was null. Elapsed time: 34ms." The Windows taskbar is visible at the bottom of the screen.

Groovy unary operator 2



The screenshot shows a Groovy IDE window titled "Main.groovy" with a file icon and the identifier "439bb38dm". The toolbar includes buttons for "AI", "NEW", "GROOVY", "RUN", and a menu icon. The main text area contains the following Groovy code:

```
1 package com.app
2 class GroovyOperatorsExample3 {
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 }
12 }
```

On the right side, the "Output" panel displays the results of the execution:

```
Output:
Unary plus = 10
Unary minus = -10
```

Groovy Assignment Operator

```
Main.groovy 439bb38dm  NEW GROOVY   
```

```
1 package com.app
2 class GroovyOperatorsExample5 {
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 }
```

Output:

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

Relational operators

```
  OneCompiler   PRICING EDITOR CHALLENGES COMPANY & MORE LOGIN
```

```
Main.groovy 439bb38dm  NEW GROOVY   
```

```
1 package com.app
2 class GroovyOperatorsExample6 {
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 "Relational Operator equals [c = a == b] ----> " + c
11        c = a != b
12        println "Relational Operator different [c = a != b] ----> " + c
13        c = a < b
14        println "Relational Operator less than [c = a < b] ----> " + c
15        c = a <= b
16        println "Relational Operator less than equal to [c = a <= b] ----> " + c
17        c = a > b
18        println "Relational Operator greater than [c = a > b] ----> " + c
19        c = a >= b
20        println "Relational Operator greater than equal to [c = a >= b] ----> " + c
21    }
22 }
23 }
```

Output:

```
a = 10
b = 12
Relational Operator equals [c = a == b] ----> false
Relational Operator different [c = a != b] ----> true
Relational Operator less than [c = a < b] ----> true
Relational Operator less than equal to [c = a <= b] ----> true
Relational Operator greater than [c = a > b] ----> false
Relational Operator greater than equal to [c = a >= b] ----> false
```

Logical operators

Example 1

Main.groovy439bb38dm

AI

NEW

GROOVY

RUN

```
1 package com.app
2 class GroovyOperatorsExample7 {
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 }
13
14
```

Output:
Logical AND operator = true
Logical OR operator = true
Logical NOT operator = true

Example 2

Main.groovy439bb38dm

AI

NEW

GROOVY

RUN

```
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 }

```

Output:
true

Bitwise operators

Main.groovy439bb38dm

AI

NEW

GROOVY

RUN

```
1 package com.app
2 class GroovyOperatorsExample10 {
3
4     static void main(args) {
5         int a = 0b00101111
6         println "a = 0b00101111 ----> "+a
7         int b = 0b000010101
8         println "b = 0b000010101 ----> "+b
9         println "(a & a) ----> "+(a & a)
10        println "(a & b) ----> "+(a & b)
11        println "(a | a) ----> "+(a | a)
12        println "(a | a) ----> "+(a | b)
13
14        int c = 0b11111111
15        println "c = 0b11111111"
16        println "((a ^ a) & c) ----> "+((a ^ a) & c)
17        println "((a ^ b) & c) ----> "+((a ^ b) & c)
18        println "((~a) & c) ----> "+((~a) & c)
19    }
20 }

```

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

Example 2

Main.groovy439bb38dm

AI

NEW

GROOVY

RUN

```
1 package com.app
2 class GroovyOperatorsExample11 {
3     static void main(args) {
4         int a = 23
5         int b = 43
6         println "Converting Integer to Binary a = 23 ----> " + Integer.toBinaryString(a)
7         println "Converting Integer to Binary b = 43 ----> " + Integer.toBinaryString(b)
8         println "Converting binary to integer 10111 ----> a = " + Integer.parseInt("10111")
9         println "Converting binary to integer 101011 ----> b = " + Integer.parseInt("101011")
10    }
11 }
```

Output:

Converting Integer to Binary a = 23 ----> 10111
Converting Integer to Binary b = 43 ----> 101011
Converting binary to integer 10111 ----> a = 23
Converting binary to integer 101011 ----> b = 23

Conditional operators

Example programs

Main.groovy439bb38dm

AI

NEW

GROOVY

RUN

```
1 package com.app
2 class GroovyOperatorsExample1 {
3     static void main(args) {
4         String Answer
5         String s = 'javatpoint'
6         println Answer = s ? 'Found' : 'Not Found'
7         println Answer = s ? 'Found'
8     }
9 }
10
```

Output:

Found
javatpoint

Main.groovy439bb38dm

AI

NEW

GROOVY

RUN

```
1 package com.app
2 class GroovyOperatorsExample12 {
3     static void main(args) {
4         println "(!true) ----> "+(!true)
5         println "(!'javatpoint') ----> "+(!'javatpoint')
6         println "(!Null) ----> "+(!'')
7     }
8 }
9
10
11
```

Output:

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