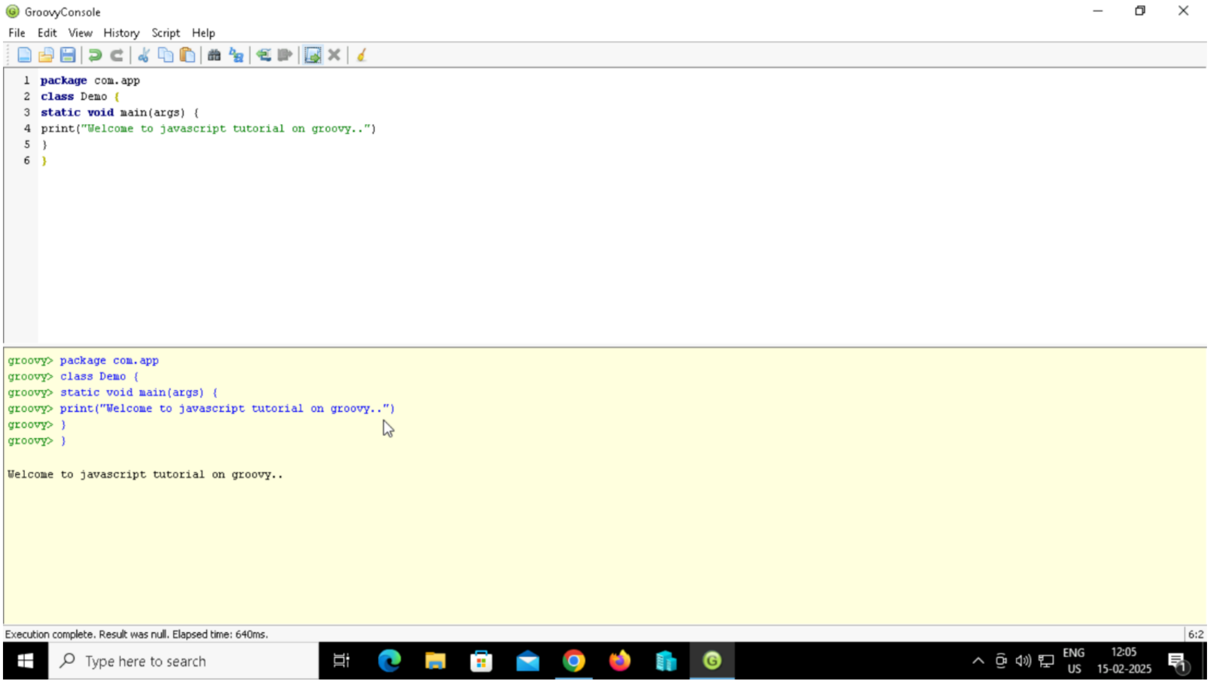
Groovy Assignment

First Groovy program



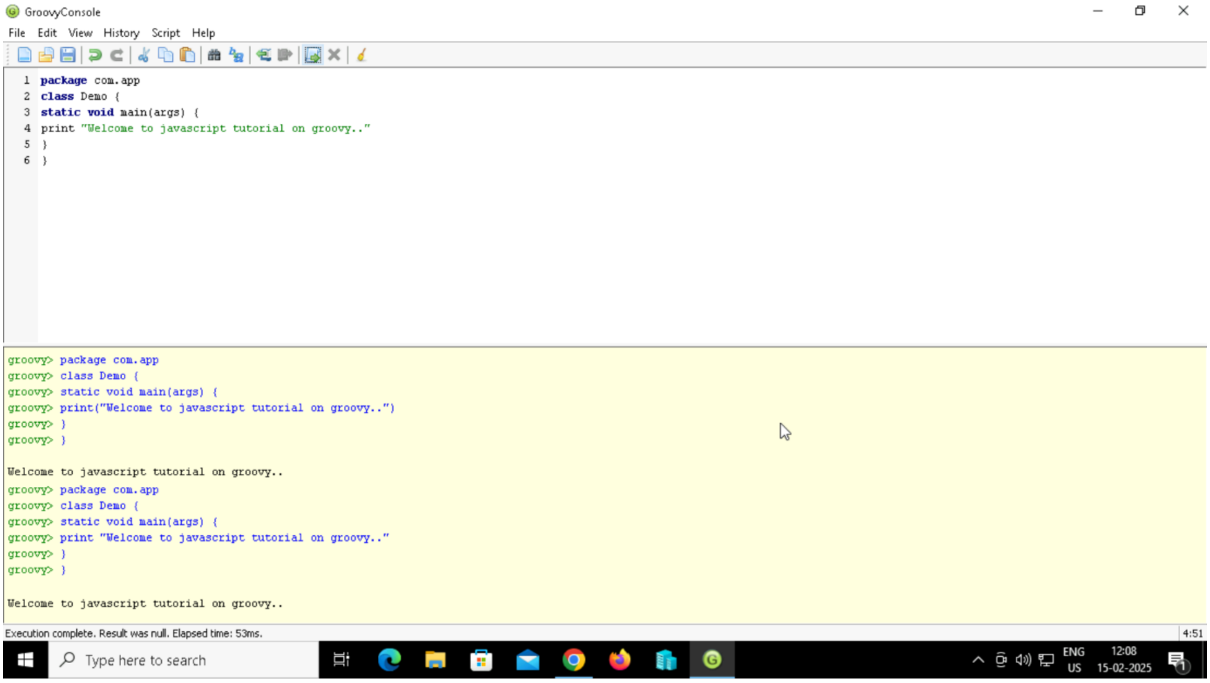
```
package com.app
class Demo {
static void main(args) {
print("Welcome to javascript tutorial on groovy..")
}
}
```

```
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..
```

Execution complete. Result was null. Elapsed time: 640ms.

Basic Syntax in groovy

Example 1

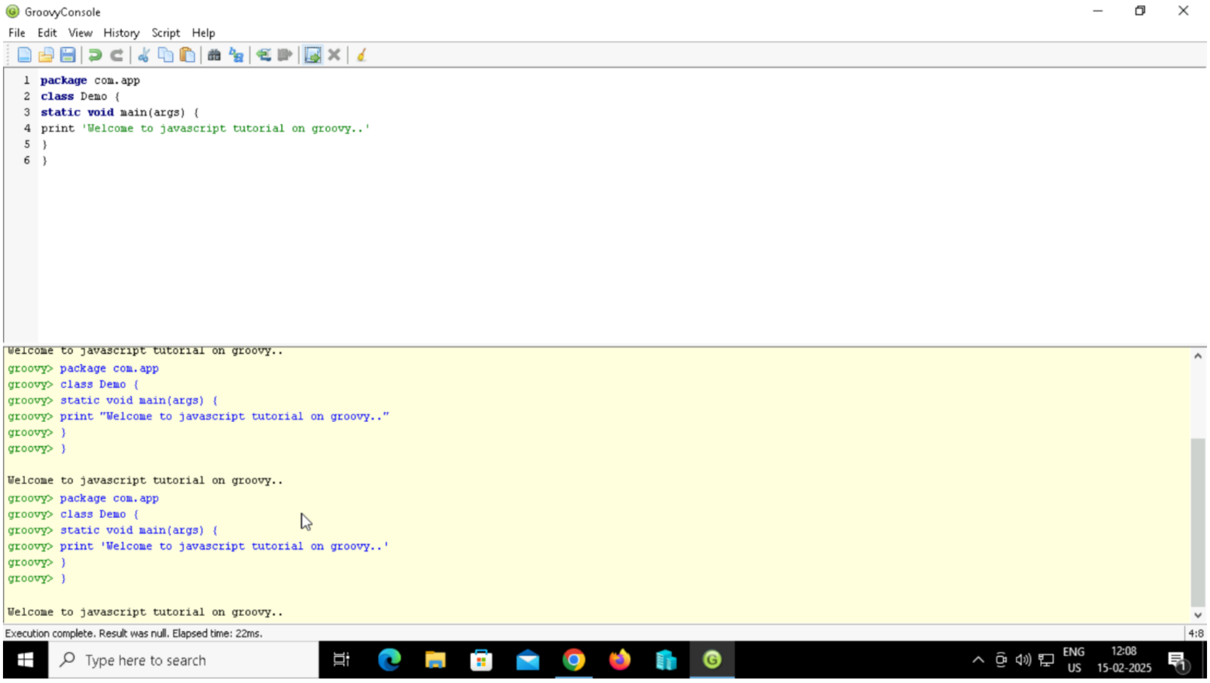


```
package com.app
class Demo {
static void main(args) {
print "Welcome to javascript tutorial on groovy.."
}
}
```

```
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..
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..
```

Execution complete. Result was null. Elapsed time: 53ms.

Example 2

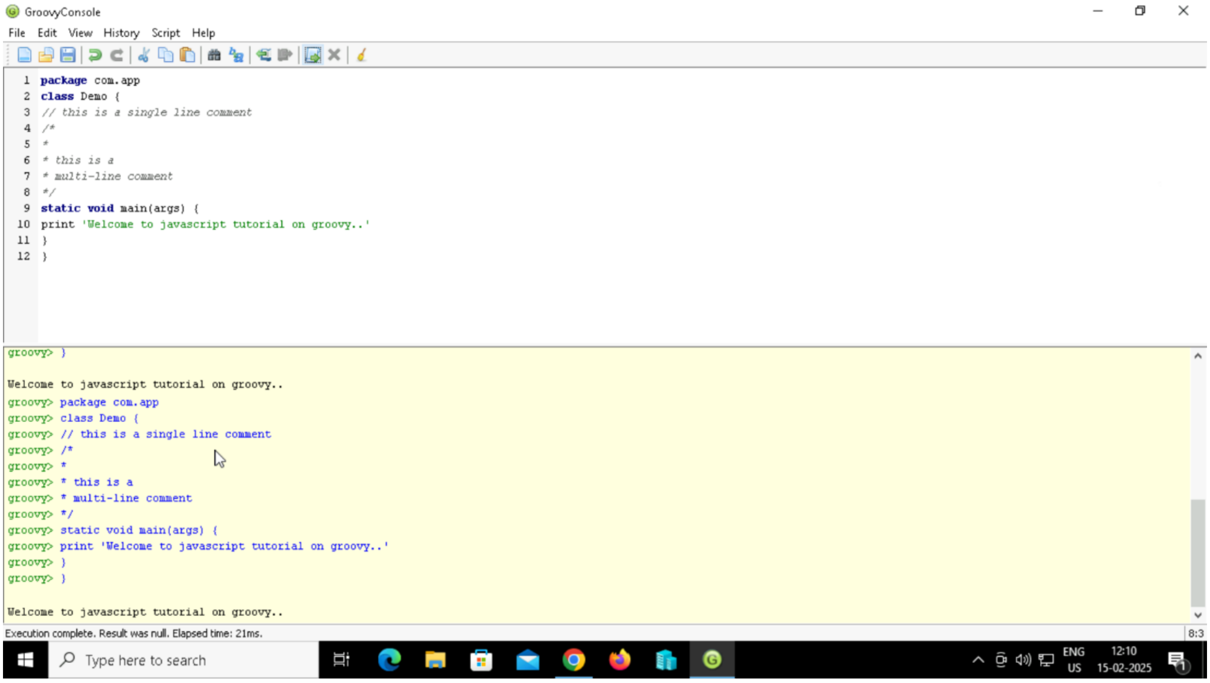


```
package com.app
class Demo {
static void main(args) {
print 'Welcome to javascript tutorial on groovy..'
}
}
```

```
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..
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..
```

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

Example 3

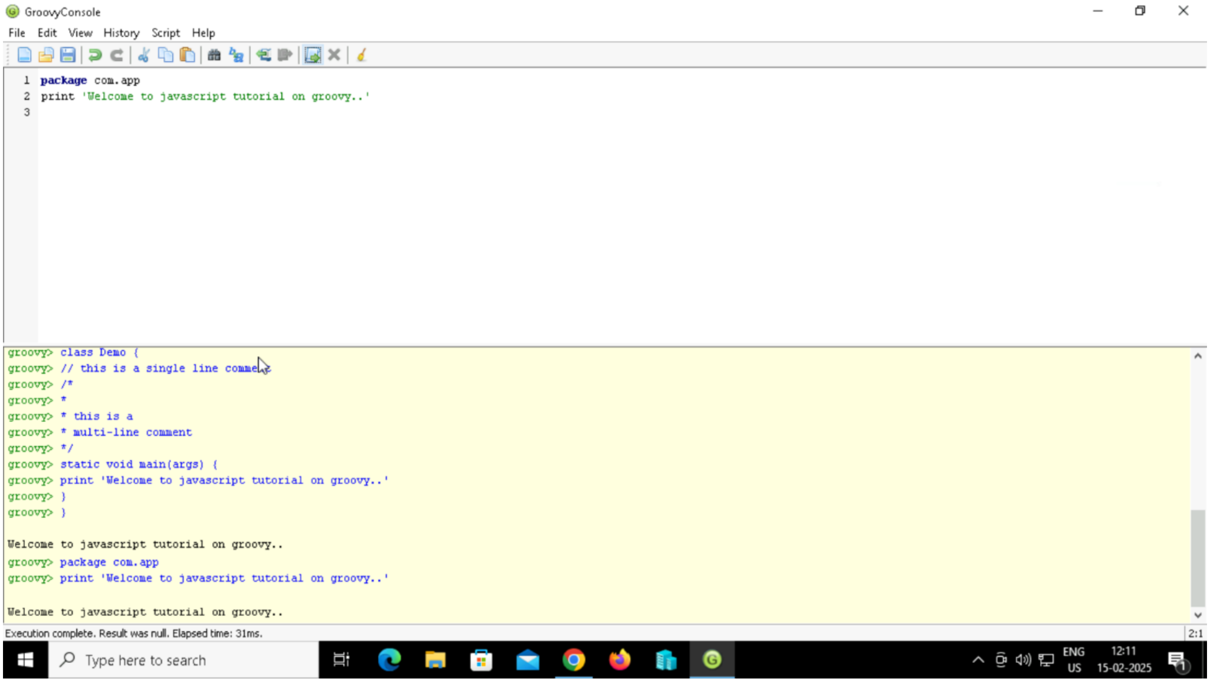


```
package com.app
class Demo {
// this is a single line comment
/*
*
* this is a
* multi-line comment
*/
static void main(args) {
print 'Welcome to javascript tutorial on groovy..'
}
}
```

```
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> }

Welcome to javascript tutorial on groovy..
```

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

Example 4



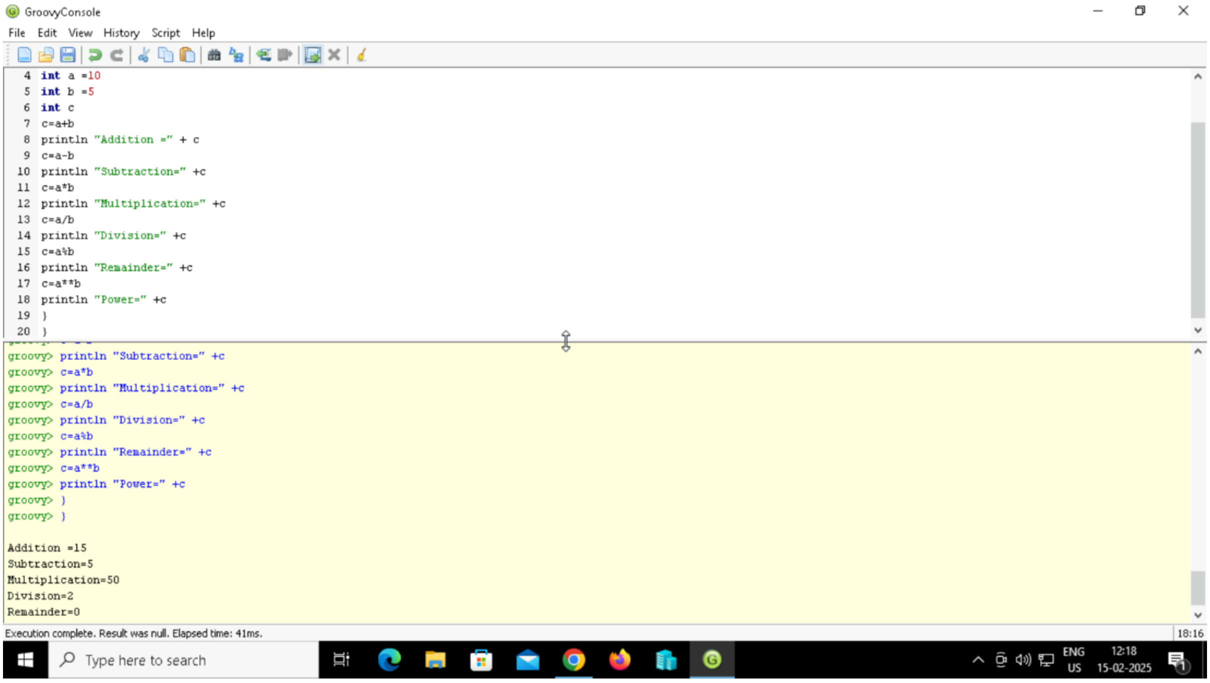
```
package com.app
print 'Welcome to javascript tutorial on groovy..'
```

```
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> }

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

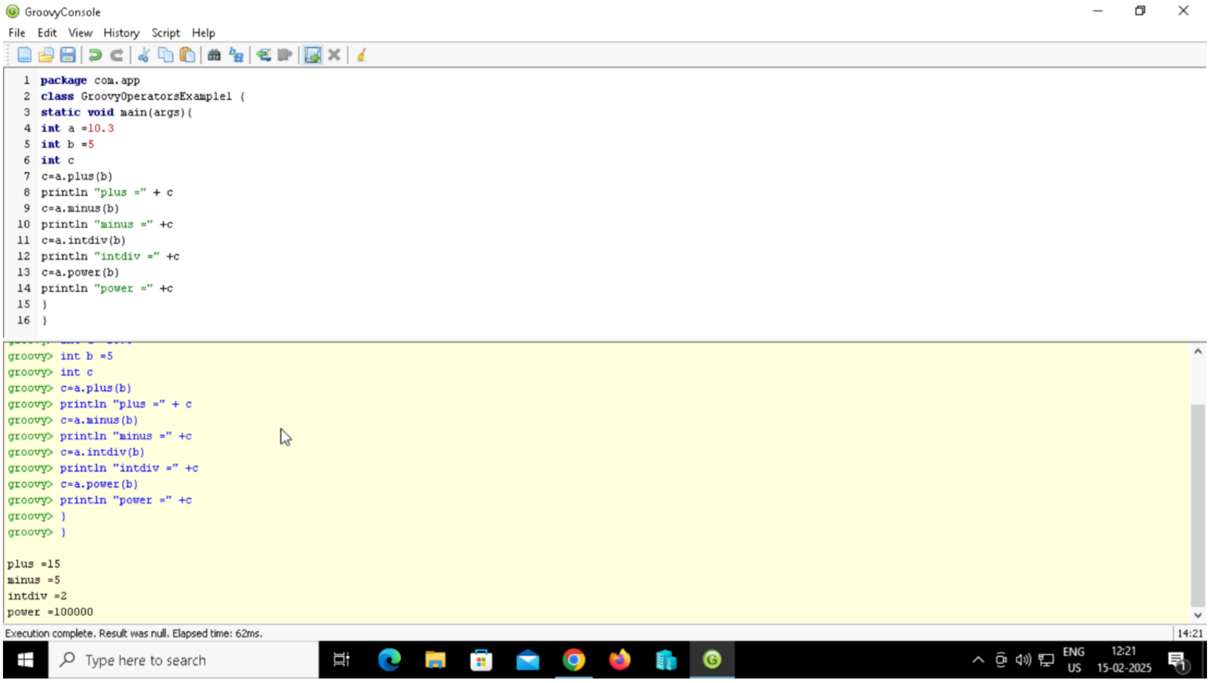


```
int a =10
int b =5
int c
c=a+b
println "Addition =" + c
c=a-b
println "Subtraction=" +c
c=a*b
println "Multiplication=" +c
c=a/b
println "Division=" +c
c=a%b
println "Remainder=" +c
c=a**b
println "Power=" +c
}
}
```

```
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



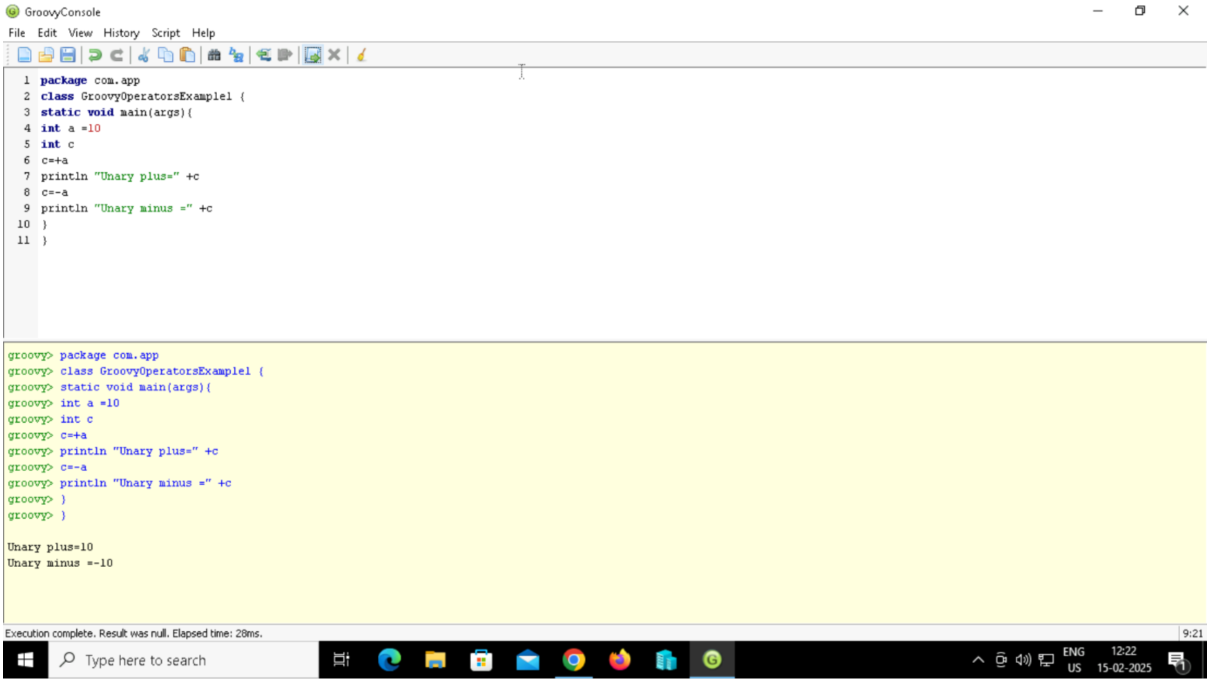
```
package com.app
class GroovyOperatorsExample1 {
static void main(args){
int a =10.3
int b =5
int c
c=a.plus(b)
println "plus =" + c
c=a.minus(b)
println "minus =" +c
c=a.intdiv(b)
println "intdiv =" +c
c=a.power(b)
println "power =" +c
}
}
```

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

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

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

Unary Operators

Example 3

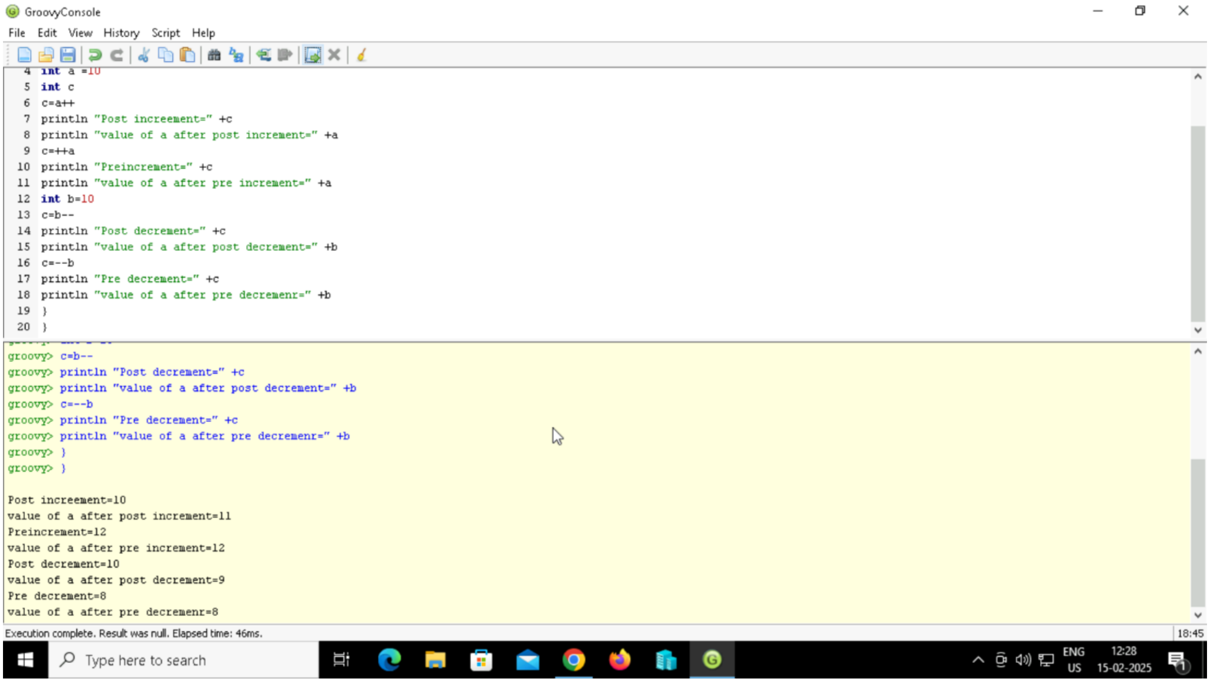


```
package com.app
class GroovyOperatorsExample1 {
static void main(args){
int a =10
int c
c=+a
println "Unary plus=" +c
c=-a
println "Unary minus =" +c
}
}
```

```
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
groovy> c=-a
groovy> println "Unary minus =" +c
groovy> }
groovy> }

Unary plus=10
Unary minus =-10
```

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

Example 4



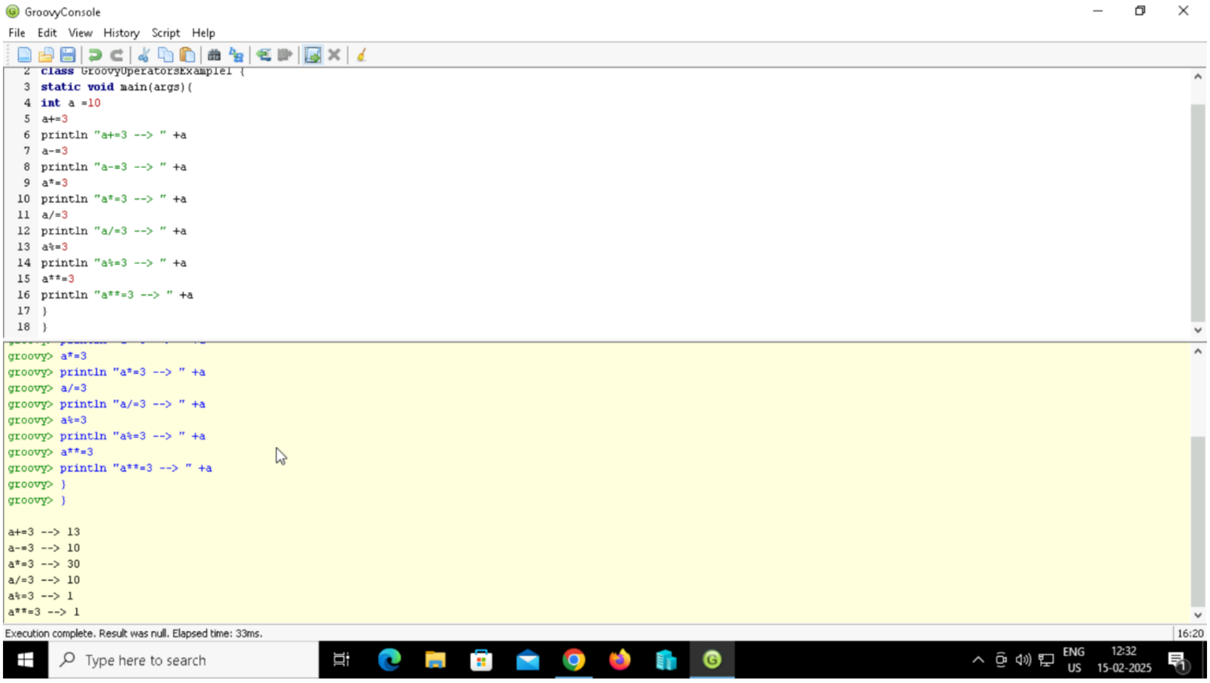
```
int a =10
int c
c=a++
println "Post increement=" +c
println "value of a after post increment=" +a
c=++a
println "Preincrement=" +c
println "value of a after pre increment=" +a
int b=10
c=b--
println "Post decrement=" +c
println "value of a after post decrement=" +b
c=--b
println "Pre decrement=" +c
println "value of a after pre decremenr=" +b
}
}
```

```
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 decremenr=" +b
groovy> }
groovy> }

Post increement=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 decremenr=8
```

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

Assignment Arithmetic Operators

Example 5



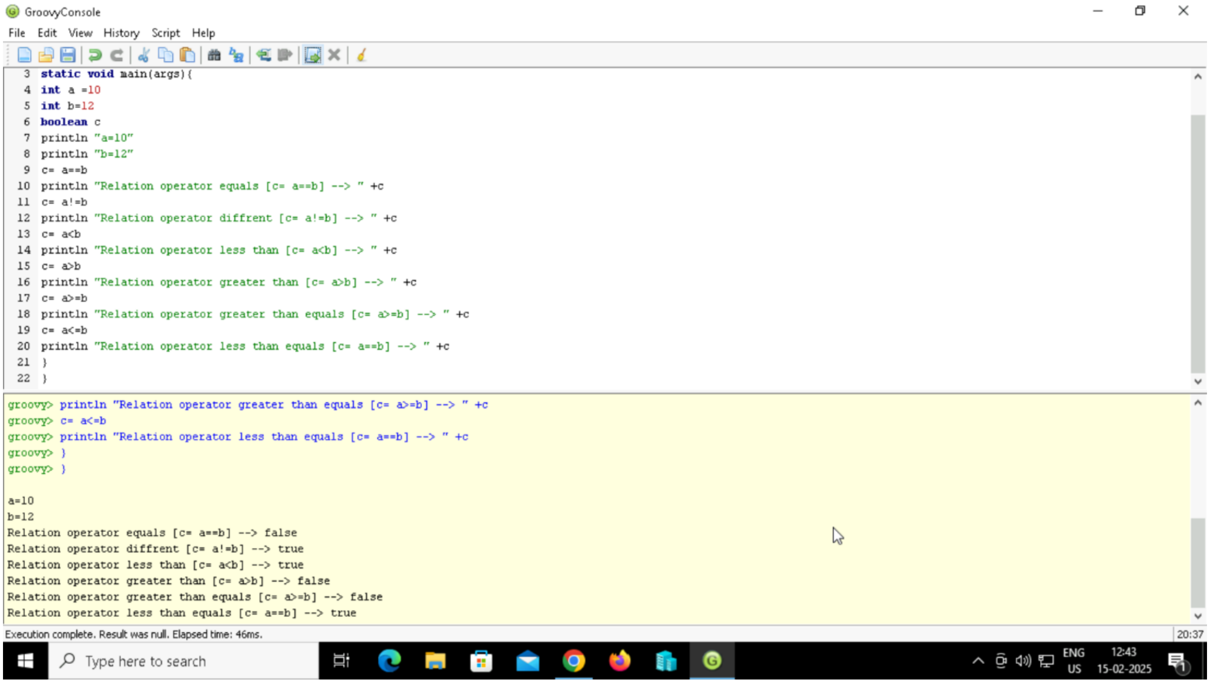
```
class GroovyOperatorsExample1 {
static void main(args){
int a =10
a+=3
println "a+=3 --> " +a
a-=3
println "a-=3 --> " +a
a*=3
println "a*=3 --> " +a
a/=3
println "a/=3 --> " +a
a%=3
println "a%=3 --> " +a
a**=3
println "a**=3 --> " +a
}
}
```

```
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> 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.

Relational Operators

Example 6



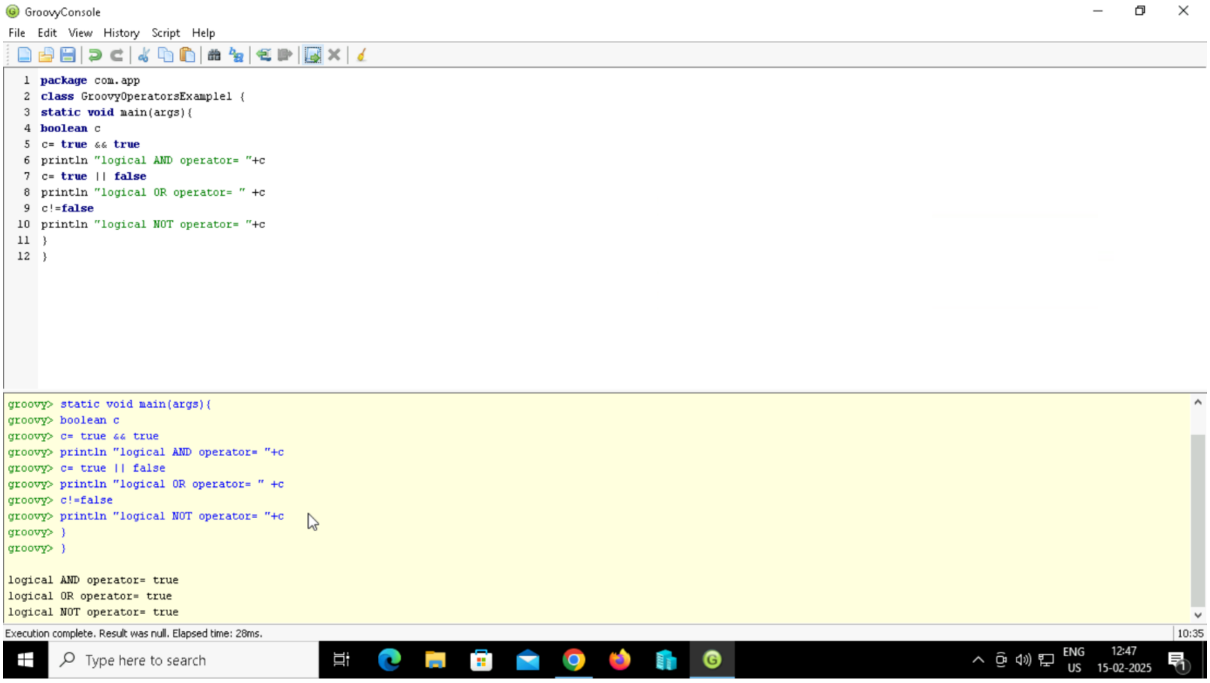
```
static void main(args){
int a =10
int b=12
boolean c
println "a=10"
println "b=12"
c= a==b
println "Relation operator equals [c= a==b] --> " +c
c= a!=b
println "Relation operator diffrent [c= a!=b] --> " +c
c= a<b
println "Relation operator less than [c= a<b] --> " +c
c= a>b
println "Relation operator greater than [c= a>b] --> " +c
c= a>=b
println "Relation operator greater than equals [c= a>=b] --> " +c
c= a<=b
println "Relation operator less than equals [c= a==b] --> " +c
}
}
```

```
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

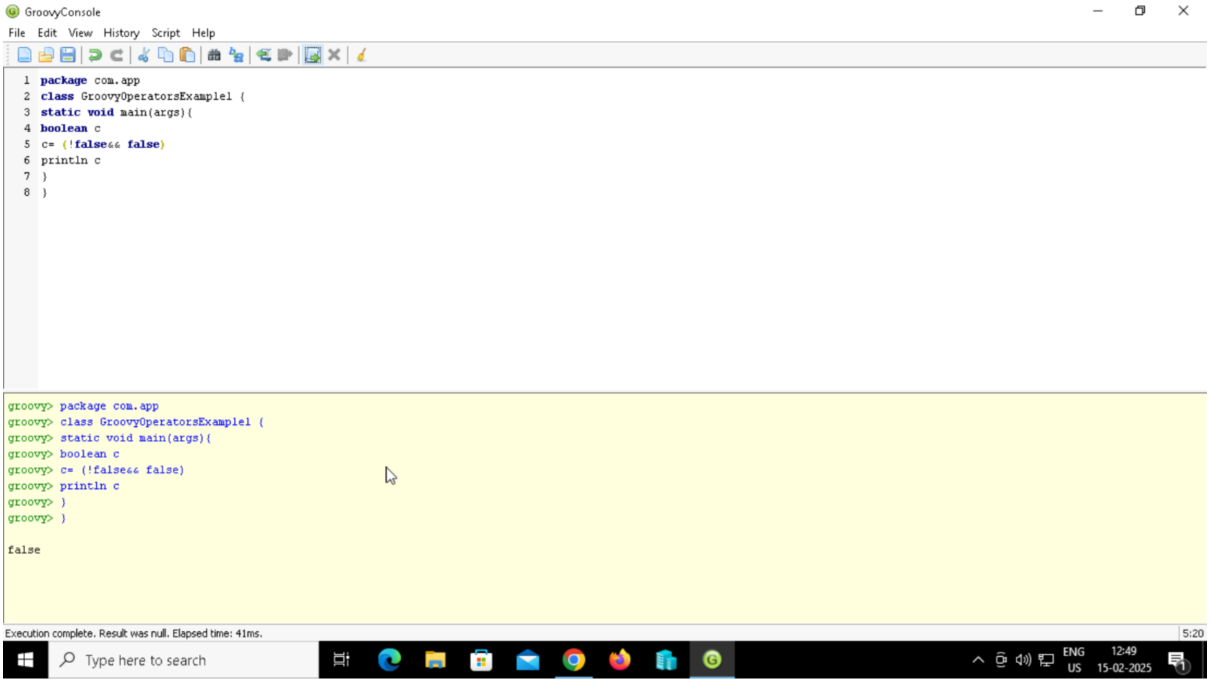


```
package com.app
class GroovyOperatorsExample1 {
static void main(args){
boolean c
c= true && true
println "logical AND operator= "+c
c= true || false
println "logical OR operator= " +c
c!=false
println "logical NOT operator= "+c
}
}
```

```
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: 28ms.

Example 8

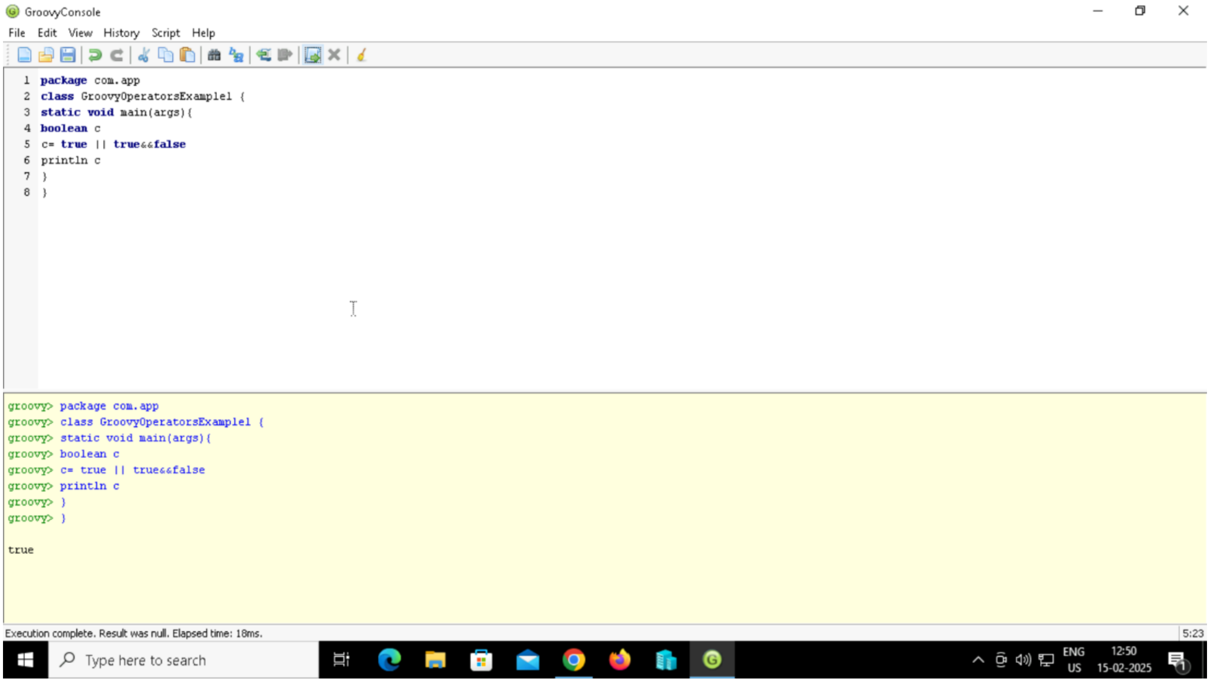


```
package com.app
class GroovyOperatorsExample1 {
static void main(args){
boolean c
c= (!false&& false)
println c
}
}
```

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

false
```

```
Execution complete. Result was null. Elapsed time: 41ms.
5:20
```

Example 9



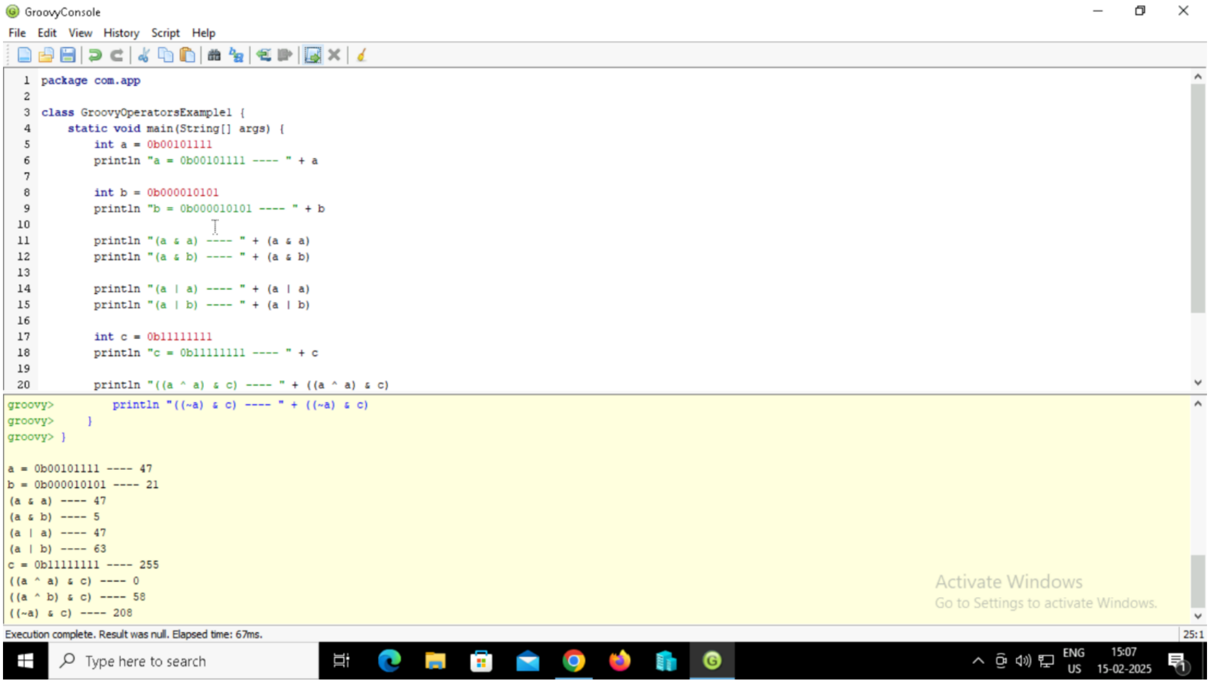
```
package com.app
class GroovyOperatorsExample1 {
static void main(args){
boolean c
c= true || true&&false
println c
}
}
```

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

true
```

```
Execution complete. Result was null. Elapsed time: 18ms.
5:23
```

Bitwise operators

Example 10

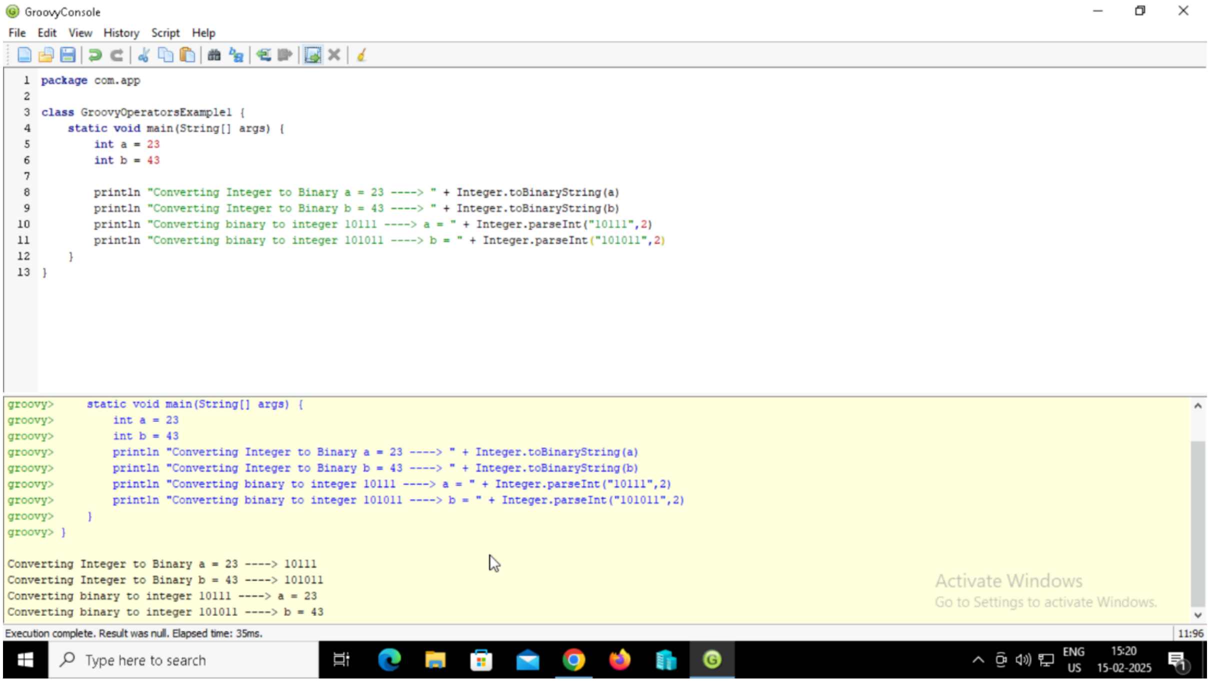


```
package com.app

class GroovyOperatorsExample1 {
    static void main(String[] args) {
        int a = 0b00101111
        println "a = 0b00101111 ---- " + a

        int b = 0b000010101
        println "b = 0b000010101 ---- " + b

        println "(a & a) ---- " + (a & a)
        println "(a & b) ---- " + (a & b)

        println "(a | a) ---- " + (a | a)
        println "(a | b) ---- " + (a | b)

        int c = 0b11111111
        println "c = 0b11111111 ---- " + c

        println "((a ^ a) & c) ---- " + ((a ^ a) & c)
```

```
groovy>         println "((~a) & c) ---- " + ((~a) & c)
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.

Example 11



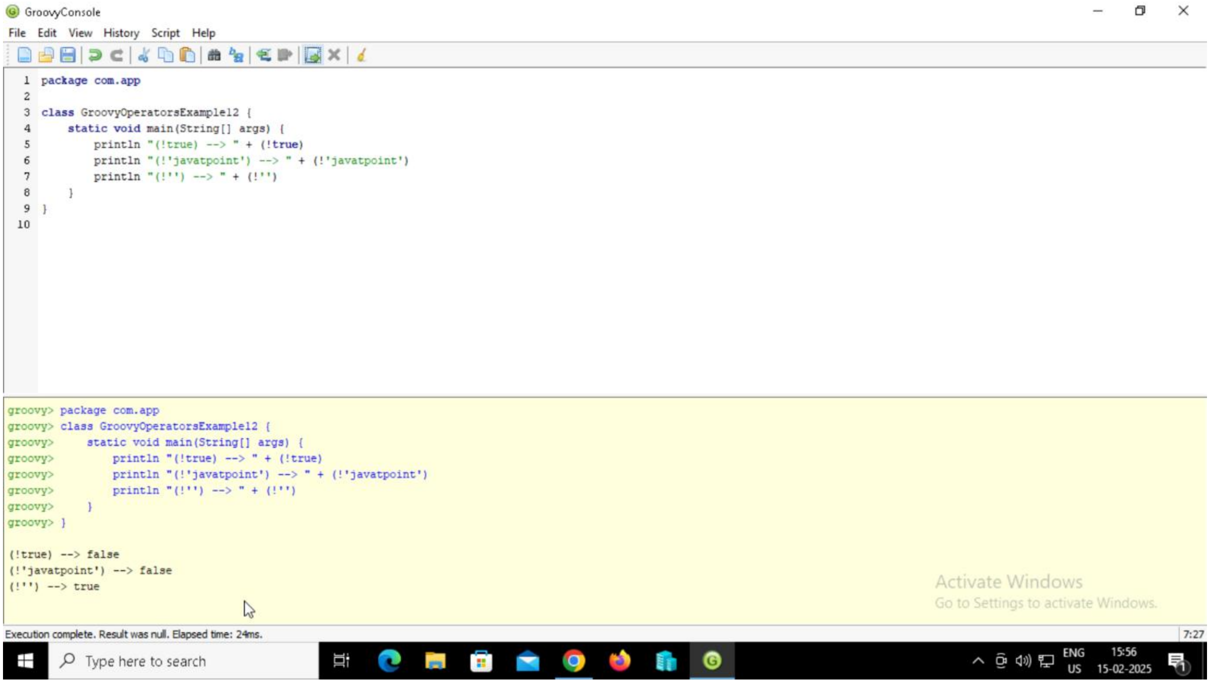
```
package com.app

class GroovyOperatorsExample1 {
    static void main(String[] args) {
        int a = 23
        int b = 43

        println "Converting Integer to Binary a = 23 ----> " + Integer.toBinaryString(a)
        println "Converting Integer to Binary b = 43 ----> " + Integer.toBinaryString(b)
        println "Converting binary to integer 10111 ----> a = " + Integer.parseInt("10111",2)
        println "Converting binary to integer 101011 ----> b = " + Integer.parseInt("101011",2)
    }
}
```

```
groovy>     static void main(String[] args) {
groovy>         int a = 23
groovy>         int b = 43
groovy>         println "Converting Integer to Binary a = 23 ----> " + Integer.toBinaryString(a)
groovy>         println "Converting Integer to Binary b = 43 ----> " + Integer.toBinaryString(b)
groovy>         println "Converting binary to integer 10111 ----> a = " + Integer.parseInt("10111",2)
groovy>         println "Converting binary to integer 101011 ----> b = " + Integer.parseInt("101011",2)
groovy>     }
groovy> }

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 = 43
```

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

Conditional operators

Example 12

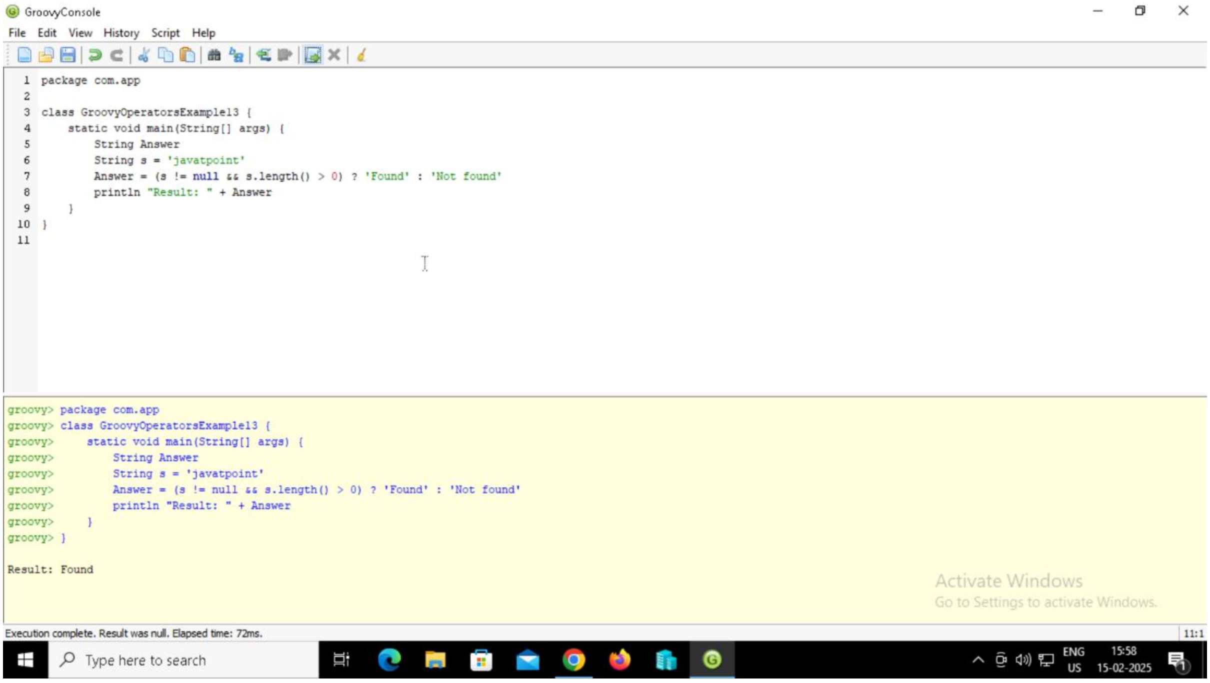


```
package com.app

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

```
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
```

Execution complete. Result was null. Elapsed time: 24ms.

Example 13

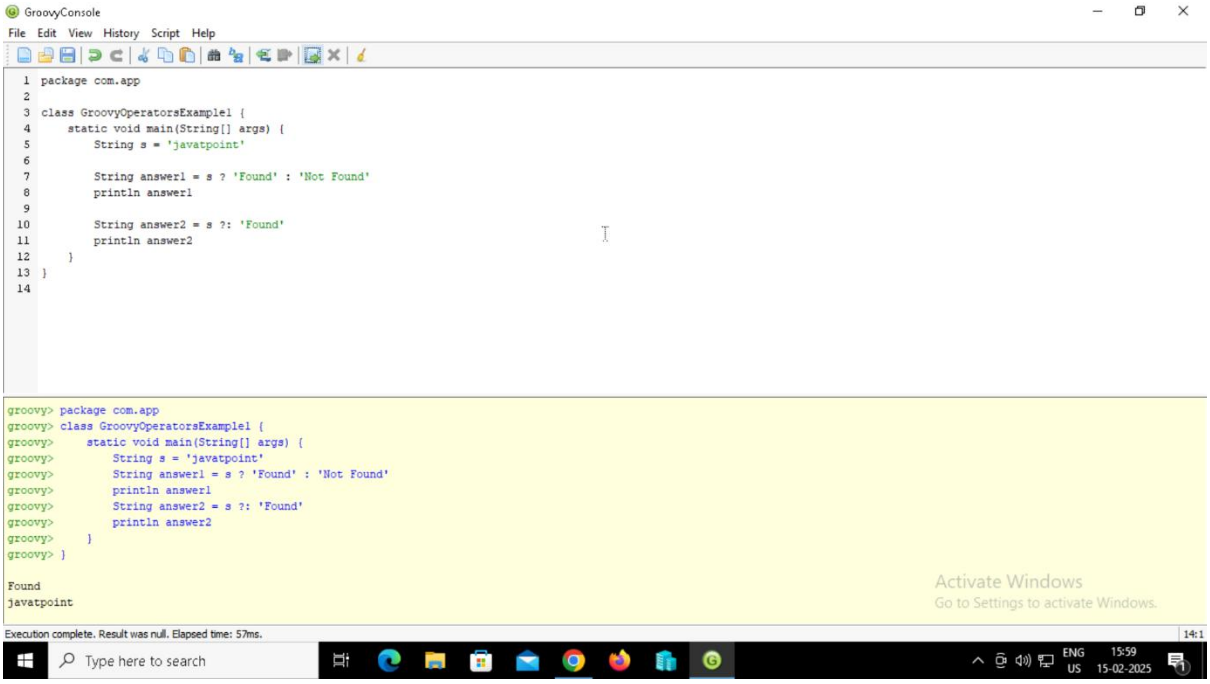


```
package com.app

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

```
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
```

Execution complete. Result was null. Elapsed time: 72ms.

Example 14



```
package com.app

class GroovyOperatorsExample1 {
    static void main(String[] args) {
        String s = 'javatpoint'

        String answer1 = s ? 'Found' : 'Not Found'
        println answer1

        String answer2 = s ?: 'Found'
        println answer2
    }
}

```

```
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
```

Execution complete. Result was null. Elapsed time: 57ms.