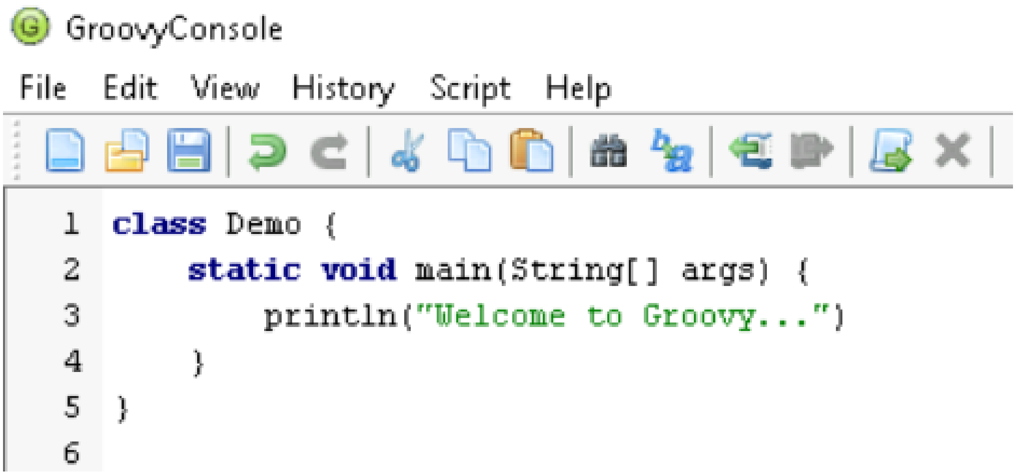
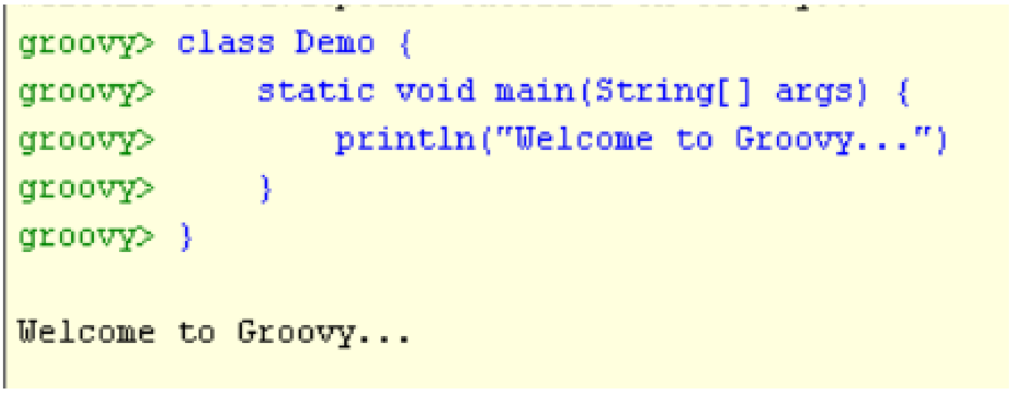
Groovy Assignment

1. **Demo**

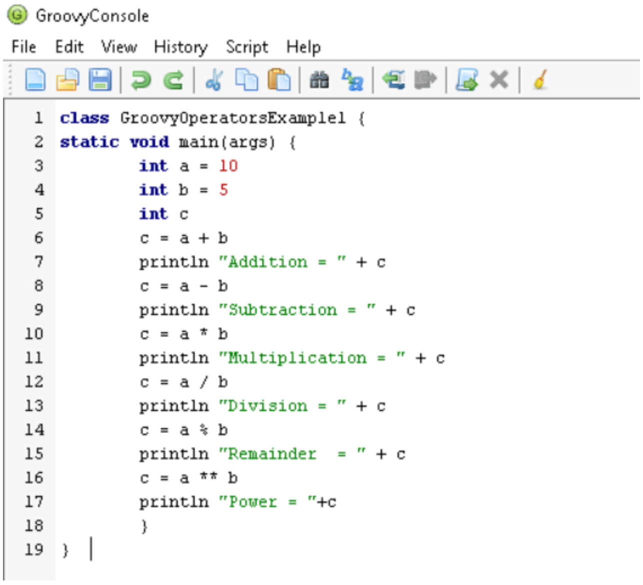


```
class Demo {
    static void main(String[] args) {
        println("Welcome to Groovy...")
    }
}
```

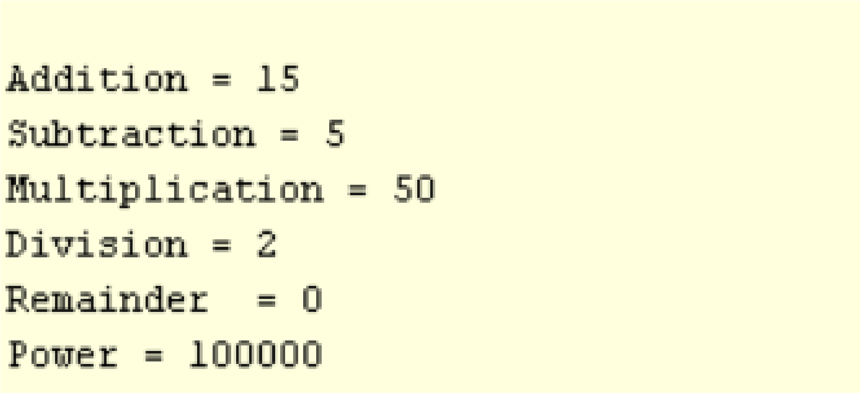


```
groovy> class Demo {
groovy>     static void main(String[] args) {
groovy>         println("Welcome to Groovy...")
groovy>     }
groovy> }

Welcome to Groovy...
```

2. **Arithmetic Operators**
    1. **Example 1:-**

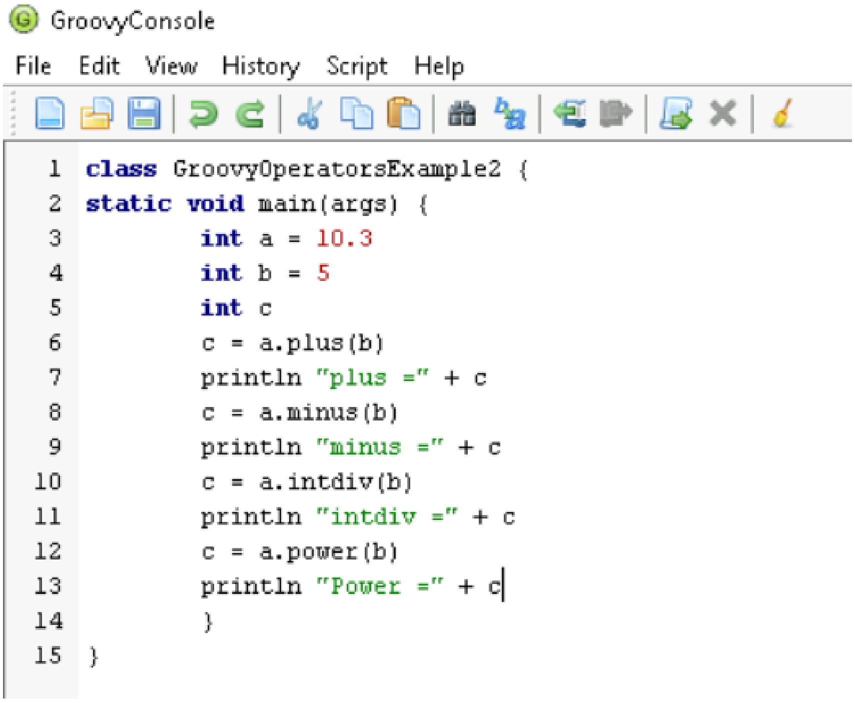


```
class GroovyOperatorsExample1 {
static void main(args) {
        int a = 10
        int b = 5
        int c
        c = a + b
        println "Addition = " + c
        c = a - b
        println "Subtraction = " + c
        c = a * b
        println "Multiplication = " + c
        c = a / b
        println "Division = " + c
        c = a % b
        println "Remainder  = " + c
        c = a ** b
        println "Power = "+c
        }
}
```

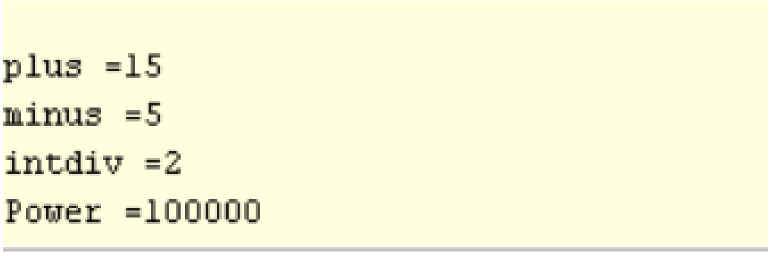


```
Addition = 15
Subtraction = 5
Multiplication = 50
Division = 2
Remainder  = 0
Power = 100000
```

2. **Example 2:-**



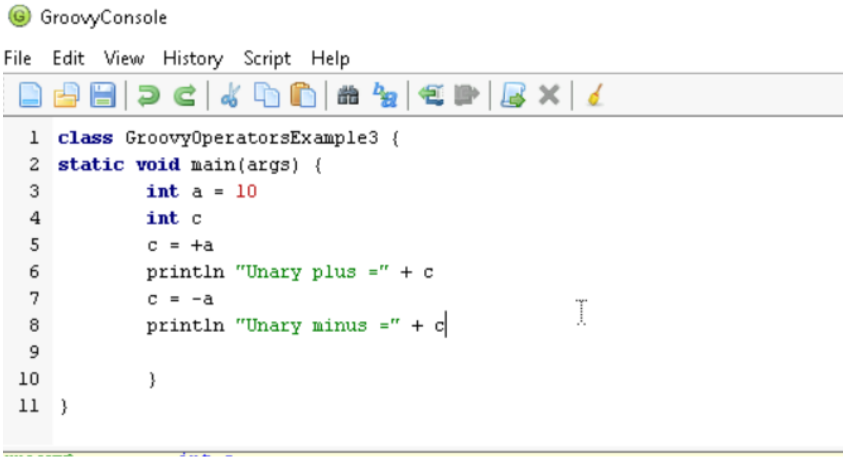
```
class GroovyOperatorsExample2 {
static void main(args) {
        int a = 10.3
        int b = 5
        int c
        c = a.plus(b)
        println "plus =" + c
        c = a.minus(b)
        println "minus =" + c
        c = a.intdiv(b)
        println "intdiv =" + c
        c = a.power(b)
        println "Power =" + c
        }
}
```



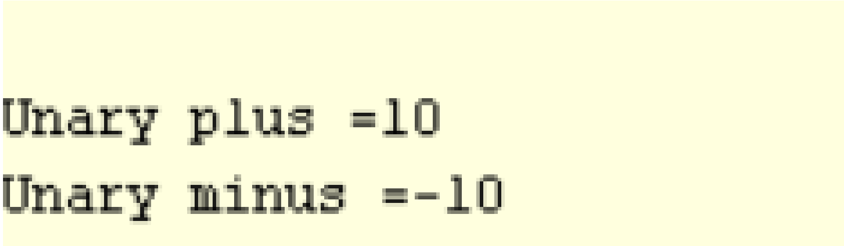
```
plus =15
minus =5
intdiv =2
Power =100000
```

3. **Unary Operators: -**

1. **Example 3**

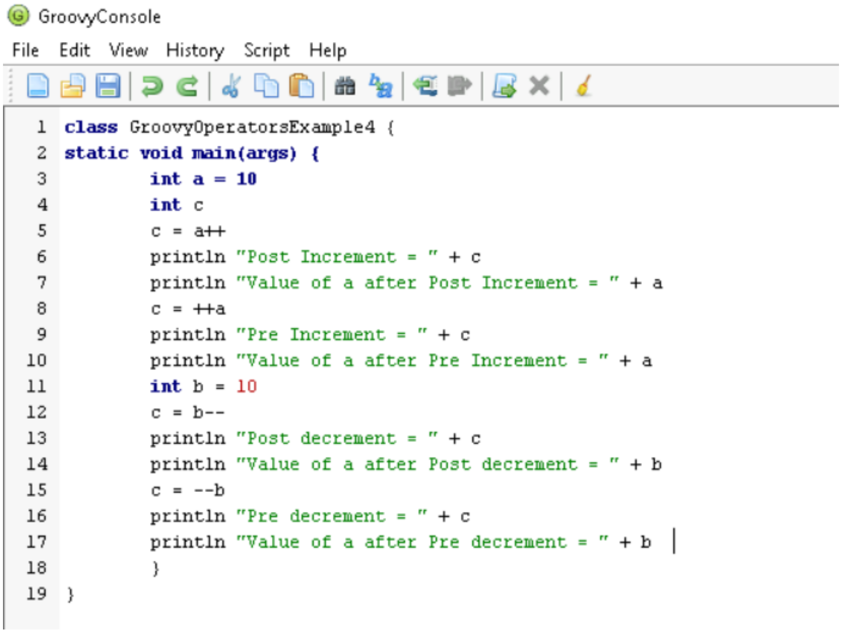


```
class GroovyOperatorsExample3 {
static void main(args) {
        int a = 10
        int c
        c = +a
        println "Unary plus =" + c
        c = -a
        println "Unary minus =" + c

        }
}
```

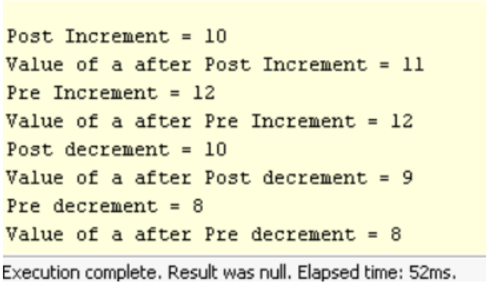


```
Unary plus =10
Unary minus =-10
```

2. **Example 4-**

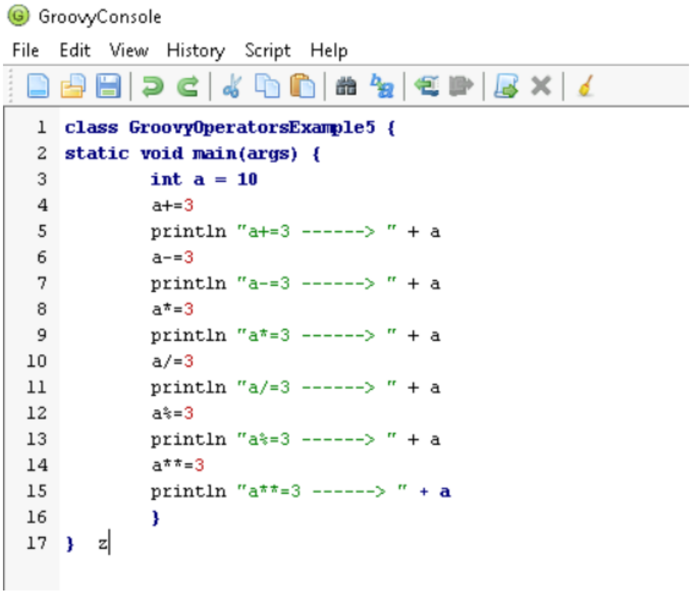


```
class GroovyOperatorsExample4 {
static void main(args) {
        int a = 10
        int c
        c = a++
        println "Post Increment = " + c
        println "Value of a after Post Increment = " + a
        c = ++a
        println "Pre Increment = " + c
        println "Value of a after Pre Increment = " + 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 decrement = " + b
        }
}
```

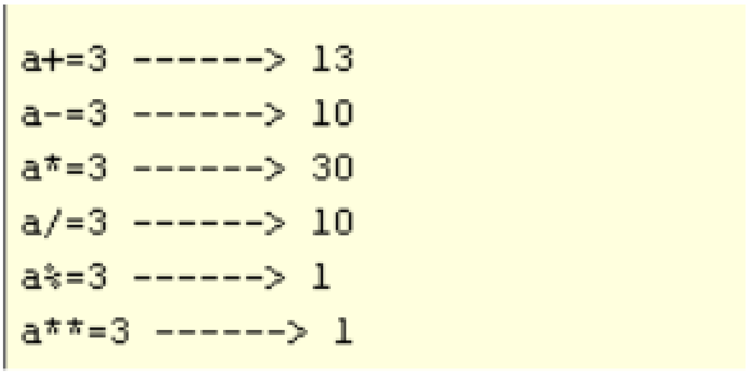


```
Post Increment = 10
Value of a after Post Increment = 11
Pre Increment = 12
Value of a after Pre Increment = 12
Post decrement = 10
Value of a after Post decrement = 9
Pre decrement = 8
Value of a after Pre decrement = 8
Execution complete. Result was null. Elapsed time: 52ms.
```

4. Assignment Arithmetic Operations: -

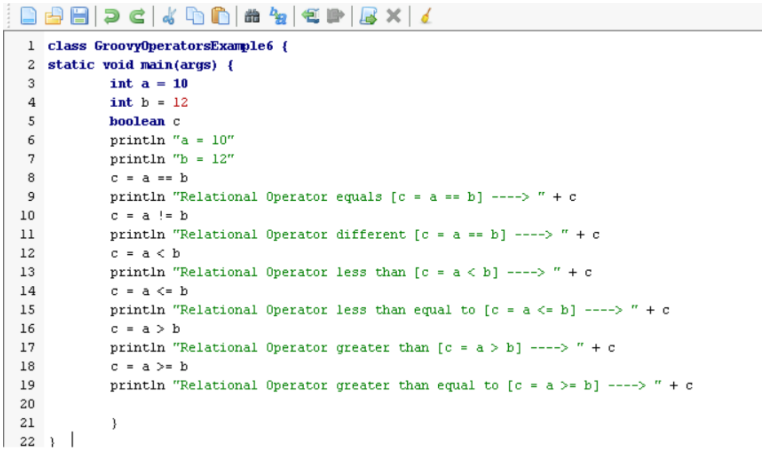


```
class GroovyOperatorsExample5 {
static void main(args) {
        int a = 10
        a+=3
        println "a+=3 ------> " + a
        a-=3
        println "a-=3 ------> " + a
        a*=3
        println "a*=3 ------> " + a
        a/=3
        println "a/=3 ------> " + a
        a%=3
        println "a%=3 ------> " + a
        a**=3
        println "a**=3 ------> " + a
        }
}  z
```

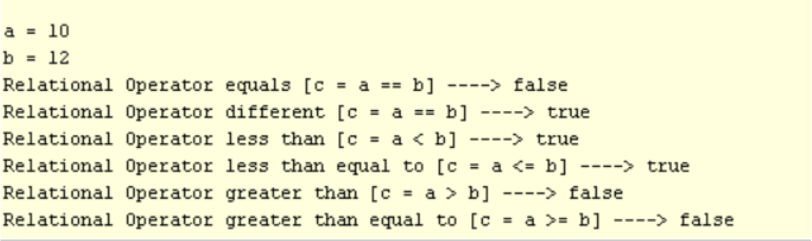


```
a+=3 ------> 13
a-=3 ------> 10
a*=3 ------> 30
a/=3 ------> 10
a%=3 ------> 1
a**=3 ------> 1
```

5. Relational Operators: -

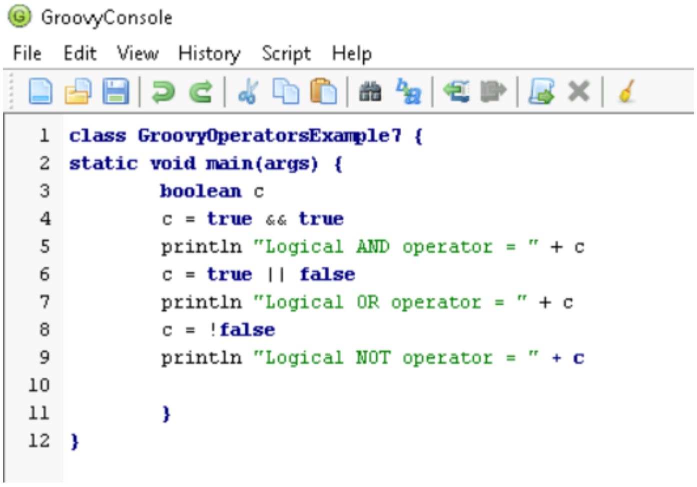


```
class GroovyOperatorsExample6 {
static void main(args) {
        int a = 10
        int b = 12
        boolean c
        println "a = 10"
        println "b = 12"
        c = a == b
        println "Relational Operator equals [c = a == b] ----> " + c
        c = a != b
        println "Relational Operator different [c = a == b] ----> " + c
        c = a < b
        println "Relational Operator less than [c = a < b] ----> " + c
        c = a <= b
        println "Relational Operator less than equal to [c = a <= b] ----> " + c
        c = a > b
        println "Relational Operator greater than [c = a > b] ----> " + c
        c = a >= b
        println "Relational Operator greater than equal to [c = a >= b] ----> " + c

        }
}
```

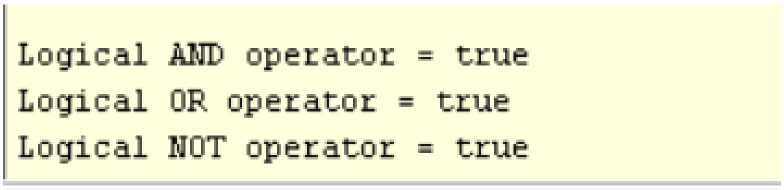


```
a = 10
b = 12
Relational Operator equals [c = a == b] ----> false
Relational Operator different [c = a == b] ----> true
Relational Operator less than [c = a < b] ----> true
Relational Operator less than equal to [c = a <= b] ----> true
Relational Operator greater than [c = a > b] ----> false
Relational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

6. Logical Operators: -

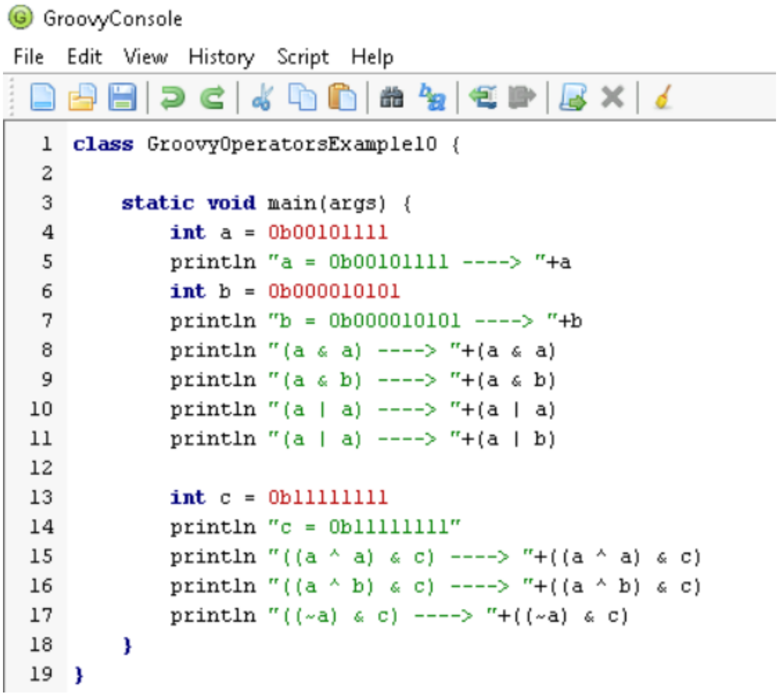


```
class GroovyOperatorsExample7 {
static void main(args) {
        boolean c
        c = true && true
        println "Logical AND operator = " + c
        c = true || false
        println "Logical OR operator = " + c
        c = !false
        println "Logical NOT operator = " + c

        }
}
```

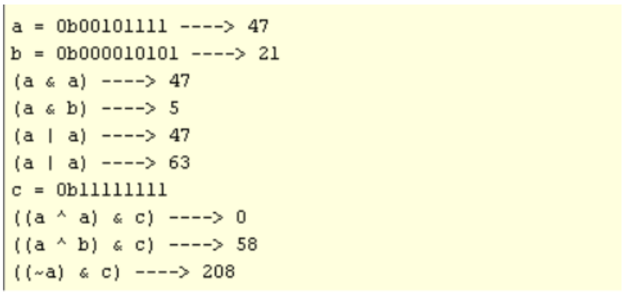


```
Logical AND operator = true
Logical OR operator = true
Logical NOT operator = true
```

7. Bitwise Operators: -

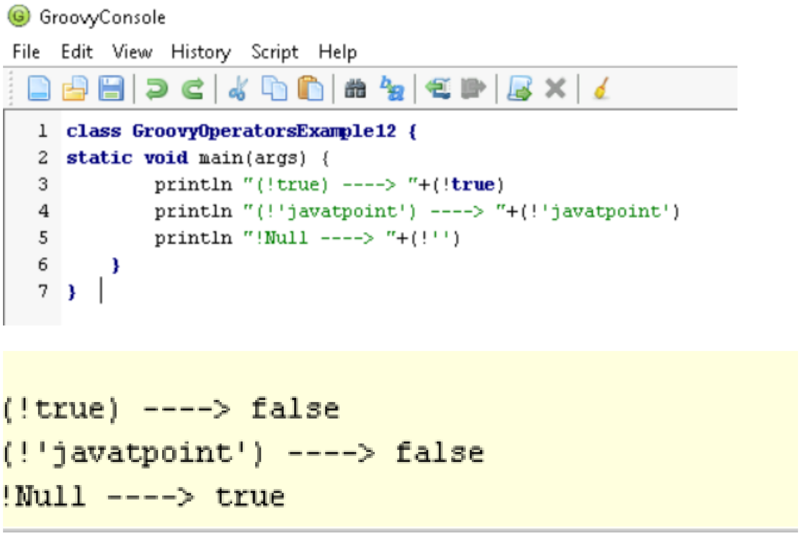


```
class GroovyOperatorsExample10 {

    static void main(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 | a) ----> "+(a | b)

        int c = 0b11111111
        println "c = 0b11111111"
        println "((a ^ a) & c) ----> "+((a ^ a) & c)
        println "((a ^ b) & c) ----> "+((a ^ b) & c)
        println "((~a) & c) ----> "+((~a) & c)
    }
}
```



```
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
```

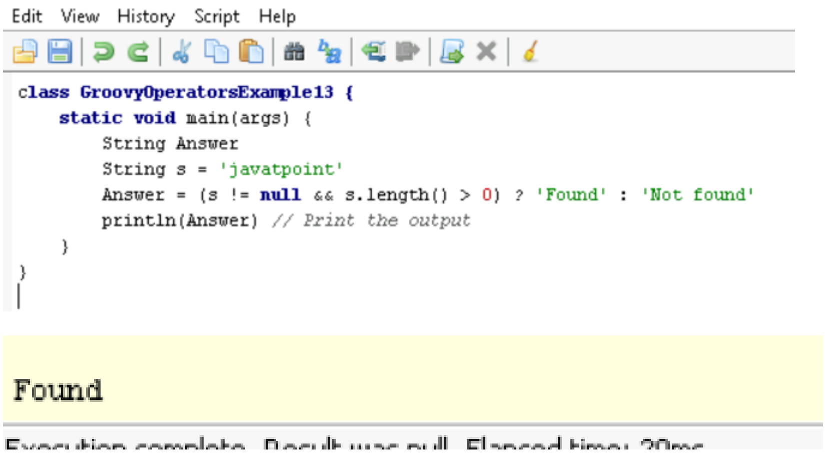
8. Conditional Operators: -



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

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

9. Ternary Operators: -

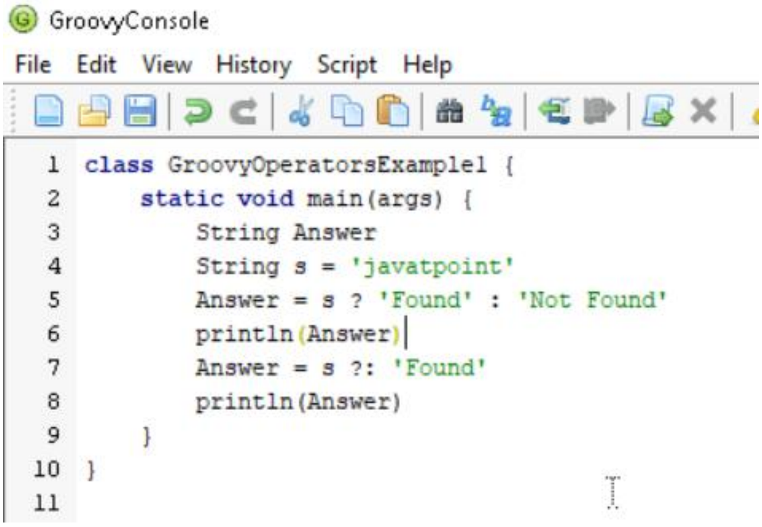


```
class GroovyOperatorsExample13 {
    static void main(args) {
        String Answer
        String s = 'javatpoint'
        Answer = (s != null && s.length() > 0) ? 'Found' : 'Not found'
        println(Answer) // Print the output
    }
}
```

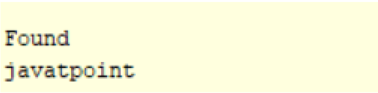
```
Found
```

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

10. Elvis Operators: -



```
class GroovyOperatorsExample1 {
    static void main(args) {
        String Answer
        String s = 'javatpoint'
        Answer = s ? 'Found' : 'Not Found'
        println(Answer)
        Answer = s ?: 'Found'
        println(Answer)
    }
}
```



```
Found
javatpoint
```