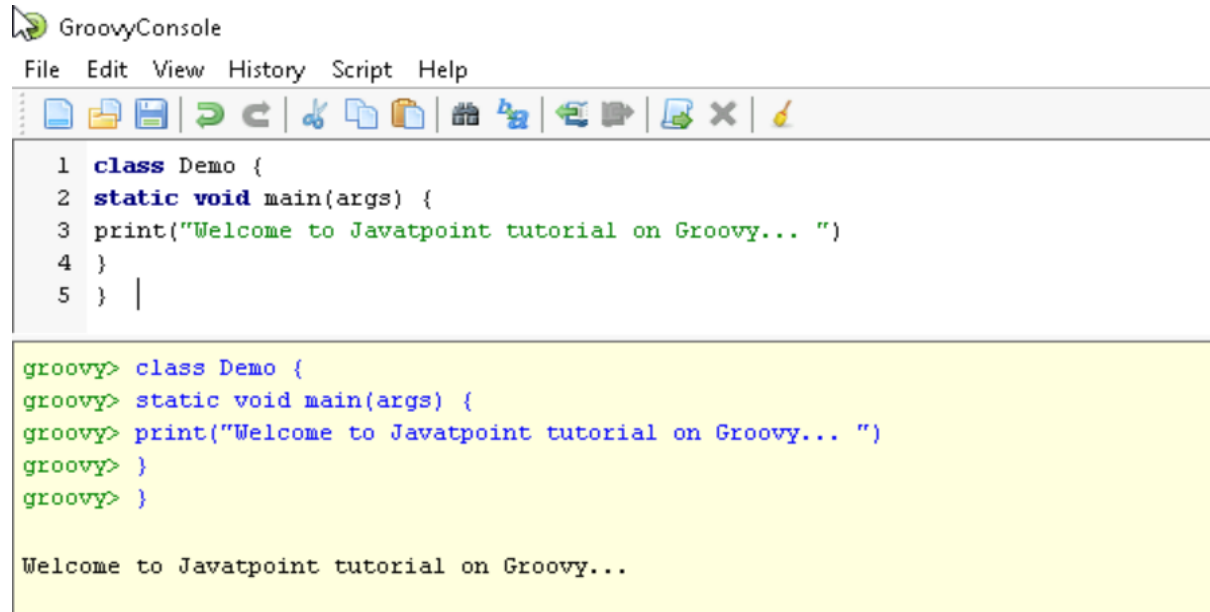


## GROOVY ASSIGNMENT

### Groovy Basic Syntax



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 (new, open, save, print), editing (undo, redo, cut, copy, paste), and execution (run, stop, refresh). The script area contains the following code:

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

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

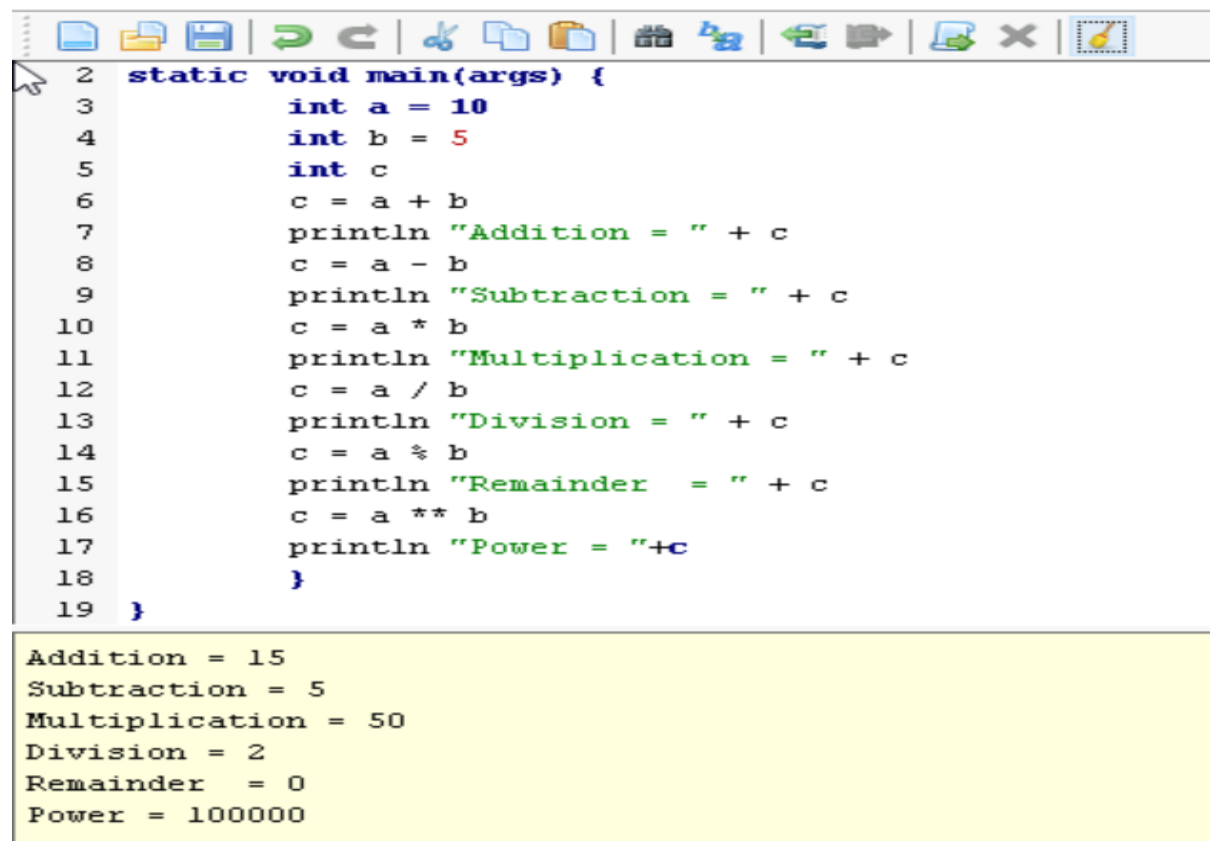
```
groovy> class Demo {
groovy> static void main(args) {
groovy> print("Welcome to Javatpoint tutorial on Groovy... ")
groovy> }
groovy> }

Welcome to Javatpoint tutorial on Groovy...
```

### GROOVY OPERATORS

In groovy, operators are symbols which are used to tell the compiler to perform specified operations. Types of operators are:

#### 1. Arithmetic operators



The screenshot shows the GroovyConsole application window with a script demonstrating arithmetic operations. The script is as follows:

```
2 static void main(args) {
3     int a = 10
4     int b = 5
5     int c
6     c = a + b
7     println "Addition = " + c
8     c = a - b
9     println "Subtraction = " + c
10    c = a * b
11    println "Multiplication = " + c
12    c = a / b
13    println "Division = " + c
14    c = a % b
15    println "Remainder = " + c
16    c = a ** b
17    println "Power = "+c
18 }
19 }
```

The output area shows the results of these operations:

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

## 2. Unary operators

```
1 class GroovyOperatorsExample4 {
2     static void main(args) {
3         int a = 10
4         int c
5         c = a++
6         println "Post Increment = " + c
7         println "Value of a after Post Increment = " + a
8         c = ++a
9         println "Pre Increment = " + c
10        println "Value of a after Pre Increment = " + a
11        int b = 10
12        c = b--
13        println "Post decrement = " + c
14        println "Value of a after Post decrement = " + b
15        c = --b
16        println "Pre decrement = " + c
17        println "Value of a after Pre decrement = " + b
18    }
19 }
```

```
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
```

## 3. Assignment arithmetic operators

```
1 class GroovyOperatorsExample5 {
2     static void main(args) {
3         int a = 10
4         a+=3
5         println "a+=3 -----> " + a
6         a-=3
7         println "a-=3 -----> " + a
8         a*=3
9         println "a*=3 -----> " + a
10        a/=3
11        println "a/=3 -----> " + a
12        a%=3
13        println "a%=3 -----> " + a
14        a**=3
15        println "a**=3 -----> " + a
16    }
17 }
```

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

## 4. Relational operators

```
1 class GroovyOperatorsExample6 {
2 static void main(args) {
3     int a = 10
4     int b = 12
5     boolean c
6     println "a = 10"
7     println "b = 12"
8     c = a == b
9     println "Relational Operator equals [c = a == b] ----> " + c
10    c = a != b
11    println "Relational Operator different [c = a == b] ----> " + c
12    c = a < b
13    println "Relational Operator less than [c = a < b] ----> " + c
14    c = a <= b
15    println "Relational Operator less than equal to [c = a <= b] ----> " + c
16    c = a > b
17    println "Relational Operator greater than [c = a > b] ----> " + c
18    c = a >= b
19    println "Relational Operator greater than equal to [c = a >= b] ----> " + c
20 }
21 }
```

```
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
```

## 5. Logical operators

```
1 class GroovyOperatorsExample7 {
2 static void main(args) {
3     boolean c
4     c = true && true
5     println "Logical AND operator = " + c
6     c = true || false
7     println "Logical OR operator = " + c
8     c = !false
9     println "Logical NOT operator = " + c
10
11 }
12 }
```

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

## 6. Bitwise operators

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

```
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
```

## 7. Conditional operators

```
1 class GroovyOperatorsExample12 {
2     static void main(args) {
3         println "(!true) ----> "+(!true)
4         println "(!'javatpoint') ----> "+(!'javatpoint')
5         println "!Null ----> "+(!'')
6     }
7 }
```

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