Thumati Pavan Venkata Narendra Kumar

EMP-ID-289219

GROOVY-ASSIGNMENTS-1

1. First Groovy Program.

1.groovy - GroovyConsole

File Edit View History Script Help

```
package com.app

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

```
groovy> package com.app
groovy> class Demo {
groovy>     static void main(String[] args) {
groovy>         println("Welcome to Javatpoint tutorial on Groovy...")
groovy>     }
groovy> }

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

2. Basic syntax in groovy.

1. Printing a line without using round brackets.

2.groovy - GroovyConsole

File Edit View History Script Help

```
1 package com.app
2
3 class Demo {
4     static void main(String[] args) {
5         println "Welcome to Javatpoint tutorial on Groovy..."
6     }
7 }
8
```

Administrator: C:\Windows\System32\cmd.exe

```
Microsoft Windows [Version 10.0.19045.4291]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator\Desktop\DevOps-Training\pavan\Groovy Home\Groovy Scripts>groovy 2.groovy
Welcome to Javatpoint tutorial on Groovy...

C:\Users\Administrator\Desktop\DevOps-Training\pavan\Groovy Home\Groovy Scripts>
```

2. Using double quotes as well as single quotes can be used in a string.

3.groovy * - GroovyConsole

File Edit View History Script Help

```
1 package com.app
2
3 class Demo {
4     static void main(String[] args) {
5         println 'Welcome to Javatpoint tutorial on Groovy...'
6     }
7 }
```

Administrator: C:\Windows\System32\cmd.exe

```
C:\Users\Administrator\Desktop\DevOps-Training\pavan\Groovy Home\Groovy Scripts>
C:\Users\Administrator\Desktop\DevOps-Training\pavan\Groovy Home\Groovy Scripts>
C:\Users\Administrator\Desktop\DevOps-Training\pavan\Groovy Home\Groovy Scripts>groovy 3.groovy
Welcome to Javatpoint tutorial on Groovy...

C:\Users\Administrator\Desktop\DevOps-Training\pavan\Groovy Home\Groovy Scripts>
```

3. **Using single line comment as well as a multi-line comment.**

4.groovy - GroovyConsole

File Edit View History Script Help

```
1  package com.app
2
3  class Demo {
4      // this is a single line comment
5      /*
6       * this is a
7       * multi-line comment
8       */
9      static void main(String[] args) {
10         println "Welcome to Javatpoint tutorial on Groovy..."
11     }
12 }
13
```

Administrator: C:\Windows\System32\cmd.exe

```
C:\Users\Administrator\Desktop\DevOps-Training\pavan\Groovy Home\Groovy Scripts>groovy 4.groovy
Welcome to Javatpoint tutorial on Groovy...

C:\Users\Administrator\Desktop\DevOps-Training\pavan\Groovy Home\Groovy Scripts>
```

4. **It is not necessary to have a class or the main function.**

5.groovy - GroovyConsole

File Edit View History Script Help

```
1  package com.app
2  println "Welcome to Javatpoint tutorial on Groovy..."
3
```

Administrator: C:\Windows\System32\cmd.exe

```
C:\Users\Administrator\Desktop\DevOps-Training\pavan\Groovy Home\Groovy Scripts>groovy 5.groovy
Welcome to Javatpoint tutorial on Groovy...

C:\Users\Administrator\Desktop\DevOps-Training\pavan\Groovy Home\Groovy Scripts>
```

3. Operators in groovy.

1. Arithmetic operators

Example 1:

arithmetic_1.groovy - GroovyConsole

File Edit View History Script Help

```
1  package com.app
2
3  class GroovyOperatorsExample1 {
4      static void main(String[] args) {
5          int a = 10
6          int b = 5
7          int c
8
9          c = a + b
10         println "Addition = " + c
11
12         c = a - b
13         println "Subtraction = " + c
14
15         c = a * b
16         println "Multiplication = " + c
17
18         c = a / b
19         println "Division = " + c
20
21         c = a % b
22         println "Remainder = " + c
23
24         c = a ** b
25         println "Power = " + c
26     }
27 }
28
```

```
groovy> package com.app
groovy> class GroovyOperatorsExample1 {
groovy>     static void main(String[] args) {
groovy>         int a = 10
groovy>         int b = 5
groovy>         int c
groovy>         c = a + b
groovy>         println "Addition = " + c
groovy>         c = a - b
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
Power = 100000
```

Example 2:

arithmetic_2.groovy - GroovyConsole

File Edit View History Script Help

```
1  package com.app
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         }
16 }
```

```
groovy> package com.app
groovy> class GroovyOperatorsExample2 {
groovy> static void main(args) {
groovy>         int a = 10.3
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
```

2. **Unary operators**

Example 1:

unary_1.groovy - 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 }
```

```
groovy> package com.app
groovy> class GroovyOperatorsExample3 {
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>         }
groovy> }

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

Example 2

unary_2.groovy - GroovyConsole

File Edit View History Script Help

```
1  package com.app
2  class GroovyOperatorsExample4 {
3  static void main(args) {
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 "Pre Increment = " + 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 }
```

```
groovy> package com.app
groovy> class GroovyOperatorsExample4 {
groovy> static void main(args) {
groovy>         int a = 10
groovy>         int c
groovy>         c = a++
groovy>         println "Post Increment = " + c
groovy>         println "Value of a after Post Increment = " + a
groovy>         c = ++a
groovy>         println "Pre Increment = " + c
groovy>         println "Value of a after Pre Increment = " + a
groovy>         int b = 10
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
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**

Example 1:

assignmentArithmetic_1.groovy - GroovyConsole

File Edit View History Script Help

```
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 }
```

```
groovy> package com.app
groovy> class GroovyOperatorsExample5 {
groovy> static void main(args) {
groovy>         int a = 10
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>         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
```

4. Relational operators

Example:

relational_1.groovy - GroovyConsole

File Edit View History Script Help

```
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         }
22 }
```

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

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**

Example 1:

logical_1.groovy - GroovyConsole

File Edit View History Script Help

```
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 }
```

```
groovy> package com.app
groovy> class GroovyOperatorsExample7 {
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>         }
groovy> }

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

Example 2:

logical _2.groovy - GroovyConsole

File Edit View History Script Help

```
1 package com.app
2 class GroovyOperatorsExample8 {
3 static void main(args) {
4         boolean c
5         c = (!false && false)
6         println  c
7     }
8 }
```

```
groovy> package com.app
groovy> class GroovyOperatorsExample8 {
groovy> static void main(args) {
groovy>         boolean c
groovy>         c = (!false && false)
groovy>         println  c
groovy>     }
groovy> }

false
```

Example 3:

logical_3.groovy - GroovyConsole

File Edit View History Script Help

```
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>     }
groovy> }

true
```

6. Bitwise operators

Example 1:

bitwise_1.groovy - GroovyConsole

File Edit View History Script Help

```
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 }
```

```
groovy> package com.app
groovy> class GroovyOperatorsExample10 {
groovy>
groovy>     static void main(args) {
groovy>         int a = 0b00101111
groovy>         println "a = 0b00101111 ----> "+a
groovy>         int b = 0b000010101
groovy>         println "b = 0b000010101 ----> "+b
groovy>         println "(a & a) ----> "+(a & a)
groovy>         println "(a & b) ----> "+(a & b)
groovy>         println "(a | a) ----> "+(a | a)
groovy>         println "(a | a) ----> "+(a | b)
groovy>
groovy>         int c = 0b11111111
groovy>         println "c = 0b11111111"
groovy>         println "((a ^ a) & c) ----> "+((a ^ a) & c)
groovy>         println "((a ^ b) & c) ----> "+((a ^ b) & 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 | a) ----> 63
c = 0b11111111
((a ^ a) & c) ----> 0
((a ^ b) & c) ----> 58
((~a) & c) ----> 208
```

Example 2:

bitwise_2.groovy - GroovyConsole

File Edit View History Script Help

```
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", 2)
9          println "Converting binary to integer 101011 ----> b = " + Integer.parseInt("10111",2)
10     }
11 }
```

```
groovy> package com.app
groovy> class GroovyOperatorsExample11 {
groovy> static void main(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("10111",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 = 23
```

7. Conditional operators

Example 1:

conditional_1.groovy - GroovyConsole

File Edit View History Script Help

```
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 }
```

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

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

Example 2:

conditional_2.groovy - GroovyConsole

File Edit View History Script Help

```
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("Answer = " + Answer)
9     }
10 }
11
```

```
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("Answer = " + Answer)
groovy>     }
groovy> }

Answer = Found
```

Example 3:

conditional_3.groovy - GroovyConsole

File Edit View History Script Help

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

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

Found
javatpoint
```

Administrator: C:\Windows\System32\cmd.exe

```
C:\Users\Administrator\Desktop\DevOps-Training\pavan\Groovy Home\Groovy Scripts\Operat
ors>groovy conditional_3.groovy
Found
javatpoint

C:\Users\Administrator\Desktop\DevOps-Training\pavan\Groovy Home\Groovy Scripts\Operat
ors>
```