**GROOVY ASSIGNMENT**

1.Number Data Types

a. What data type is the number 2? How about 20? 200? Keep adding zeros and watch the data type change until it reaches BigInteger. Then do the same for 2.0.

Program:

def num1=2

println "$num1-"+num1.getClass().getName()

//integers

def num1=2

println "$num1-"+num1.getClass().getName()

def num2=200

println "$num2-"+num2.getClass().getName()

def num3=2000

println "$num3-"+num3.getClass().getName()

def num4=20000

println "$num4-"+num4.getClass().getName()

def num5=200000

println "$num5-"+num5.getClass().getName()

def num6=20000000000

println "$num6-"+num6.getClass().getName()

def num7=20000000000000000000

println "$num7-"+num7.getClass().getName()

// for float

def num8=2.0

println "$num8-"+num8.getClass().getName()

def num9=2.000

println "$num9-"+num9.getClass().getName()

def num10=2.00000000

println "$num10-"+num10.getClass().getName()

def num11=2.000000000000

println "$num11-"+num11.getClass().getName()

Answer:

PS C:\Users\user\Desktop\groovy> groovy test.groovy

2-java.lang.Integer

200-java.lang.Integer

2000-java.lang.Integer

20000-java.lang.Integer

200000-java.lang.Integer

20000000000-java.lang.Long

20000000000000000000-java.math.BigInteger

2.0-java.math.BigDecimal

2.000-java.math.BigDecimal

2.00000000-java.math.BigDecimal

2.000000000000-java.math.BigDecimal

-----------------------------------------------------------------------------------------------------------------

b. Declare a variable x of type def and assign it the sum of 1 and 1.5. What is the resulting data type?

Program:

def x = 1+2

println "$x-"+x.getClass().getName()

Answer:

PS C:\Users\user\Desktop\groovy> groovy Number\_data\_types\_b.groovy

2.5-java.math.BigDecimal

-----------------------------------------------------------------------------------------------------------------

c. What do you get when you divide 5 by 2? What is the resulting data type? If you wanted to do integer division (no remainder), what method would you call?

Program:

x=5

y=2

z=5/2

r=x.intdiv(y)

println "$x + $y = $z -->"+z.getClass().getName()

println "$x + $y = $r -->"+r.getClass().getName() //integer division

Answer:

PS C:\Users\user\Desktop\groovy> groovy Number\_datatypes\_c.groovy

5 + 2 = 2.5 -->java.math.BigDecimal

5 + 2 = 2 -->java.lang.Integer

-----------------------------------------------------------------------------------------------------------------

2. Wrapper Classes From the associated wrapper classes, find the min and max values for the Java primitives: byte, short, int, long, float, double.

Program:

//BYTE RANGE

println "Byte range starts from --> "+Byte.MIN\_VALUE

println "Byte range ends at --> "+Byte.MAX\_VALUE

//SHORT RANGE

println "Short range starts from --> "+Short.MIN\_VALUE

println "Short range starts ends at --> "+Short.MAX\_VALUE

//INTEGER RANGE

println "Integer range starts from --> "+Integer.MIN\_VALUE

println "Integer range ends at --> "+Integer.MAX\_VALUE

//LONG RANGE

println "Long range starts from --> "+Long.MIN\_VALUE

println "Long range ends at --> "+Long.MAX\_VALUE

//FLOAT RANGE

println "Float range starts from --> "+Float.MIN\_VALUE

println "Float range ends at --> "+Float.MAX\_VALUE

//DOUBLE RANGE

println "Double range starts from --> "+Double.MIN\_VALUE

println "Double range ends at --> "+Double.MAX\_VALUE

Answer:

PS C:\Users\user\Desktop\groovy> groovy 2.Wrapple\_Classes.groovy

Byte range starts from --> -128

Byte range ends at --> 127

Short range starts from --> -32768

Short range starts ends at --> 32767

Integer range starts from --> -2147483648

Integer range ends at --> 2147483647

Long range starts from --> -9223372036854775808

Long range ends at --> 9223372036854775807

Float range starts from --> 1.4E-45

Float range ends at --> 3.4028235E38

Double range starts from --> 4.9E-324

Double range ends at --> 1.7976931348623157E308

-----------------------------------------------------------------------------------------------------------------

3. 2s Complement

Create a byte variable with its maximum value. What do you get when you add 1 to it?

Program:

Byte b=127

Byte r=b+1

println "maximum byte value is $b adding 1 to it results -> $r"

Answer:

PS C:\Users\user\Desktop\groovy> groovy 3.complement.groovy

maximum byte value is 127 adding 1 to it results -> -128

-----------------------------------------------------------------------------------------------------------------

4. Strings and GroovyStrings

a. How many characters are in the string "Hello, Groovy!"?

Program:

//length of string"Hello,Groovy!"

println "Hello,Groovy!".size()

Answer:

PS C:\Users\user\Desktop\groovy> groovy test.groovy

13

-----------------------------------------------------------------------------------------------------------------

b. Define a string variable containing a name. Print a hello statement with your name using string concatenation, then using a Groovy string.

Program:

String name="Mahesh"

println "Hello $name"

println "hello "+name

println "Hello ".concat("mahesh")

Answer:

PS C:\Users\user\Desktop\groovy> groovy 4.Strings\_b.groovy

Hello Mahesh

hello Mahesh

Hello Mahesh

-----------------------------------------------------------------------------------------------------------------

c. Demonstrate that "racecar" is a palindrome by comparing it to its reverse. Do the same with "Bob", removing case sensitivity first.

Program:

String i="racecar"

String j=i.reverse()

println i.equals(j)

/\*if (i==j)

{

println "yes it's a palindrome $i == $j"

}

//print j

\*/

String str="Bob"

String str2=str.reverse()

println str.equalsIgnoreCase(str2)

Answer:

PS C:\Users\user\Desktop\groovy> groovy 4.Strings\_c.groovy

true

true

-----------------------------------------------------------------------------------------------------------------

d. Define a string variable containing the sentence, "Hello, World. How are you?". Split the sentence into an array using the split method. Count the number of words. Do the same using the tokenize method.

Program:

String str="Hello, World. How are you?"

ar=str.split()

println "length is "+ar.size()

tk=str.tokenize()

println tk

println tk.size()

Answer:

PS C:\Users\user\Desktop\groovy> groovy 4.Strings\_d.groovy

length is 5

[Hello,, World., How, are, you?]

5

-----------------------------------------------------------------------------------------------------------------

e. Using the same sentence, use array notation (square brackets) to print the substring "World".

Program:

String str="Hello, World. How are you?"

ar=str.split()

println ar[1]

Answer:

PS C:\Users\user\Desktop\groovy> groovy 4.Strings\_e.groovy

World.

-----------------------------------------------------------------------------------------------------------------

f. Use array notation to print the last word, but reversed.

Program:

String str="Hello, World. How are you?"

ar=str.split()

println ar[-1].reverse()

Answer:

PS C:\Users\user\Desktop\groovy> groovy 4.Strings\_f.groovy

?uoy

-----------------------------------------------------------------------------------------------------------------

5. Prime Numbers

Write a method called isPrime that takes an integer argument and returns a boolean. Determine whether the number is prime by dividing it by all numbers from 2 up to one less than the number.

That limit is too high, of course. How high do you have to check to be sure whether you've gone far enough?

Program:

def Prime(num){

i=2

count=0

while(i<num){

if (num%i==0){

count=count+1

if (count==1){

return false

}

}

i=i+1

}

return true

}

print "enter the num: "

num=System.console().readLine()

number=num as Integer

result=Prime(number)

if (result){

println "yes $num prime number"

}

else{ println "$num is not a primenumber"}

Answer:

PS C:\Users\user\Desktop\groovy> groovy 5.prime.groovy

enter the num: 61

yes 61 prime number

PS C:\Users\user\Desktop\groovy> groovy 5.prime.groovy

enter the num: 69

69 is not a primenumber

-----------------------------------------------------------------------------------------------------------------

1. Sorting Strings

Create a list of strings. Sort them alphabetically. Sort them by length. Sort them by length in descending order.

Advanced: Sort by length, then sort equal length strings alphabetically

Program:

def lst=["apple","banana","pineapple","mango","cheery","apples"]

println "original list --> "+lst

println "sorted list in ascending order--> "+lst.sort()

println "sorted list in descending order --> "+lst.sort().reverse()

Answer:

PS C:\Users\user\Desktop\groovy> groovy Sorting\_strings.groovy

original list --> [apple, banana, pineapple, mango, cheery, apples]

sorted list in ascending order--> [apple, apples, banana, cheery, mango, pineapple]

sorted list in descending order --> [pineapple, mango, cheery, banana, apples, apple]

-----------------------------------------------------------------------------------------------------------------

2. Processing a list of numbers

Create a list of numbers. Add them together. First double each number, then add them up. Compute their average.

Program:

def lst=[1,2,3,4,5,6,7,8,9]

def lst2=lst.collect{item -> item\*2}

sum1=0

sum2=0

for (Integer i=0;i<lst.size();i++){

sum1=sum1+lst[i]

sum2=sum2+lst2[i]

}

avg1=sum1/lst.size()

avg2=sum2/lst.size()

println "$avg1 $avg2"

Answer:

PS C:\Users\user\Desktop\groovy> groovy test.groovy

5 10

-----------------------------------------------------------------------------------------------------------------

3. Closures as a filter

Create a list of numbers. Print all elements greater than zero.

Program:

def lst=[-1,-2,3,4,5,0,6,7,8,9]

def closu=lst.findAll{item -> item >0}

println closu

Answer:

PS C:\Users\user\Desktop\groovy> groovy Closure.groovy

[3, 4, 5, 6, 7, 8, 9]

-----------------------------------------------------------------------------------------------------------------

4. Multi-line strings

Make a multi-line string. Compute the number of vowels on each line.

Program:

def str=""" hello this is groovy a programming

language which is similar to java python c

syntax.

This is the line for assignment."""

//def lst=["a","e","i","o","u","A","E","I","O","U"]

st=str.toList()

count=0

for (i in str){

if (i=='a'|| i=='e' || i=='i'||i=='o'||i=='u'){

count=count+1

}

}

println count

Answer:

PS C:\Users\user\Desktop\groovy> groovy multi\_line.groovy

33

-----------------------------------------------------------------------------------------------------------------

5. Padded binary output

Print the numbers from 0 to 15 in binary (use Java's Integer.toBinaryString() method). Use a method in String from the Groovy JDK to make all the output values have four digits.

Program:

Integer num=16

for (int i=0;i<num;i++){

k=Integer.toBinaryString(i)

println(k.padLeft(4,'0'))

}

Answer:

PS C:\Users\user\Desktop\groovy> groovy Padding\_binary

0000

0001

0010

0011

0100

0101

0110

0111

1000

1001

1010

1011

1100

1101

1110

1111

-----------------------------------------------------------------------------------------------------------------

1. Encode and decode

i. Create two strings, one for a username and one for a password. Concatenate them together, separated by a colon. Use a method from the Groovy JDK to convert the resulting String to a byte array. Then use the encodeBase64 method on byte array to create an encoded string.

ii. Decode the string by using the decodeBase64 method, and using the result as an argument to the String constructor. Use the split method to return the original username and password.

Program:

print "enter the user name: "

def user\_name=System.console().readLine()

print "enter the password: "

def password=System.console().readLine()

def id=user\_name+":"+password

//println id

byte[] encoded=Base64.getEncoder().encode(id.getBytes())

println "Encoded bytes data is --> $encoded"

j=new String(Base64.getDecoder().decode(encoded))

words=j.split(":")

user=words[0]

pssd=words[1]

println "Decoded User\_name --> $user || Password -->$pssd"

Answer:

PS C:\Users\user\Desktop\groovy> groovy Encode\_decode.groovy

enter the user name: mahesh123

enter the password: 123mahesh

Encoded bytes data is --> [98, 87, 70, 111, 90, 88, 78, 111, 77, 84, 73, 122, 79, 106, 69, 121, 77, 50, 49, 104, 97, 71, 86, 122, 97, 65, 61, 61]

Decoded User\_name --> mahesh123 || Password -->123mahesh

-----------------------------------------------------------------------------------------------------------------

2. Sorting a list

Create a class called Course, with a String attribute called name and an int attribute called days. Create a list of four course instances, where at least two have the same number of days. Sort the list by number of days. Then, sort the list by days, but when the days are equal, sort by name.

Program:

class Course {

String name

int days

def getvalues(String n ,int d)

{

def name = n

def days = d

def l = [name:n,days:d]

}

}

Course First=new Course()

Course Second=new Course()

Course Third=new Course()

Course Fourth=new Course()

def common = [First.getvalues('git', 2), Second.getvalues('groovy', 2), Third.getvalues('python', 4), Fourth.getvalues('unix', 4)]

def sort = common.sort{a, b -> b["days"] <=> a["days"] ?: a["name"] <=> b["name"]}

sort.each { println it }

Answer:

PS C:\Users\user\Desktop\groovy> groovy class\_string.groovy

[name:python, days:4]

[name:unix, days:4]

[name:git, days:2]

[name:groovy, days:2]

-----------------------------------------------------------------------------------------------------------------

3. Operator overloading

i. Create a class called Money with a double amount and a String currency (like USD or EUR). Implement a plus method that checks that the currencies are the same and, if so, returns a new Money instance with the sum of the amounts and the correct currency. Write a similar minus method.

ii. Write a MoneyTest class in Groovy that uses + and - and verifies that they work properly.