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.

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

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**:

//a//

def a = 2

println a

println a.getClass()

def b = 20

println b

println b.getClass()

def c = 200

println c

println c.getClass()

def d = 200000000000

println d

println d.getClass()

def a1 = 2.0

println a1

println a1.getClass()

//b//

def x = 1 + 1.5

println x

println x.getClass()

//c//

def y = 5 / 2

println y

println y.getClass()

def id = 5.intdiv(2)

println id

println id.getClass()

**OUTPUT**:

2

class java.lang.Integer

20

class java.lang.Integer

200

class java.lang.Integer

200000000000

class java.lang.Long

2.0

class java.math.BigDecimal

2.5

class java.math.BigDecimal

2.5

class java.math.BigDecimal

2

class java.lang.Integer

1. *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 b = 10

println b

println Byte.MIN\_VALUE

println Byte.MAX\_VALUE

short s = 100

println s

println Short.MIN\_VALUE

println Short.MAX\_VALUE

int i = 1000

println i

println Integer.MIN\_VALUE

println Integer.MAX\_VALUE

long l = 100000

println l

println Long.MIN\_VALUE

println Long.MAX\_VALUE

float f = 100.002F

println f

println Float.MIN\_VALUE

println Float.MAX\_VALUE

double d = 1000.002D

println d

println Double.MIN\_VALUE

println Double.MAX\_VALUE

**OUTPUT:**

10

-128

127

100

-32768

32767

1000

-2147483648

2147483647

100000

-9223372036854775808

9223372036854775807

100.002

1.4E-45

3.4028235E38

1000.002

4.9E-324

1.7976931348623157E308

1. *2s Complement*

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

**PROGRAM:**

byte b = 127

println b.getClass()

add = b+1

println add.getClass()

**OUTPUT:**

class java.lang.Byte

class java.lang.Integer

1. *Strings and GroovyStrings*
   1. How many characters are in the string "Hello, Groovy!"?

**PROGRAM:**

//a

string = "Hello,Groovy"

println ("string length is : " +string.length() )

**OUTPUT:**

string length is : 12

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

**PROGRAM:**

//b

def myclouser = { name -> println "Hello $name" }

myclouser.call("poorva")

**OUTPUT:**

Hello poorva

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

**PROGRAM:**

import java.util.Scanner;

public class number {

public static void main(String[] args) {

String reverse="";

System.*out*.println("Enter the string");

Scanner sc=new Scanner(System.*in*);

String original= sc.nextLine();

for(int i= original.length()-1;i>=0;i--) {

reverse=reverse+original.charAt(i);

}

if(original.equals(reverse)){

System.*out*.println("given string is palindrome");

}else{

System.*out*.println("given string is not palindrome");

}

}

**OUTPUT:**

Enter the string

racecar

given string is not palindrome

Enter the string

bob

given string is palindrome

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.

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

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

**PROGRAM:**

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

println str.split(" ")

println str.length()

println str.tokenize()

println str[7..12]

def str1 = "you"

println str1 .reverse()

**OUTPUT:**

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

26

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

World.

Uoy

1. *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:**

print "Enter the Number: "

def num = System.console().readLine().toInteger()

boolean flag = false;

for (int i = 2; i <= num / 2; ++i) {

// condition for nonprime number

if (num % i == 0) {

flag = true;

break;

}

}

if (!flag) {

println(num + " is a prime number.");}

else {

println(num + " is not a prime number.");

}

**OUTPUT:**

Enter the number: 11

11 is a prime number.

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 name = ["App", "O", "Graaapes"]

println name.sort()

//sort by length//

words = name.sort { it.size() }

println words

//sort by length in descending order//

word = name.sort { -it.size() }

println word

**OUTPUT:**

[App, Graaapes, O]

[O, App, Graaapes]

[Graaapes, App, O]

1. *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 list = [4,5,8,9]

println list.collect{it\*2}

println list.average()

**OUTPUT:**

[8, 10, 16, 18]

6.5

1. *Reading a web page*

Using the Groovy JDK, access your home page and display the source code. Print the length of each line of the home page.

**PROGRAM:**

String filepath = "F:/1st groovy project/src/com/test/demo/text1"

File myfile = new File(filepath)

println myfile.text

println myfile.size()

**OUTPUT:**

line1

line2

line3

line4

26

1. *Closures as a filter*

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

**PROGRAM:**

def list = [1,-1 ,2 ,-2]

println list.findAll { item -> item > 0 }

**OUTPUT:**

[1, 2]

1. *Multi-line strings*

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

**PROGRAM:**

def str1 = "The Java String is immutable which means it cannot be change.";

int count = 0

for(vowel in str1) {

if(vowel == 'a' || vowel == 'e' || vowel =='i' || vowel =='o' || vowel =='u' || vowel == 'A' || vowel == 'E' || vowel =='I' || vowel =='O' || vowel =='U') {

count++

}

}

println "Vowels is " +count

**OUTPUT:**

Vowels is 18

1. *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:**

int l = 0;

println("Binary is 000" +Integer.toBinaryString(l));

int m = 1;

println("Binary is 000" +Integer.toBinaryString(m));

int n = 2;

println("Binary is 00" +Integer.toBinaryString(n));

int o = 3;

println("Binary is 00" +Integer.toBinaryString(o));

int p = 4;

println("Binary is 0" +Integer.toBinaryString(p));

int q = 5;

println("Binary is 0" +Integer.toBinaryString(q));

int r = 6;

println("Binary is 0" +Integer.toBinaryString(r));

int s = 7;

println("Binary is 0" +Integer.toBinaryString(s));

int t = 8;

println("Binary is " +Integer.toBinaryString(t));

int u = 9;

println("Binary is " +Integer.toBinaryString(u));

int v = 10;

println("Binary is " +Integer.toBinaryString(v));

int w = 11;

println("Binary is " +Integer.toBinaryString(w));

int x = 12;

println("Binary is " +Integer.toBinaryString(x));

int a = 13;

println("Binary is " +Integer.toBinaryString(a));

int z = 14;

println("Binary is " +Integer.toBinaryString(z));

int b = 15;

println("Binary is " +Integer.toBinaryString(b));

**OUTPUT:**

Binary is 0000

Binary is 0001

Binary is 0010

Binary is 0011

Binary is 0100

Binary is 0101

Binary is 0110

Binary is 0111

Binary is 1000

Binary is 1001

1. *Encode and decode*
2. 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.
3. 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:**

String uname = "devops"

String password = "Groovy"

List end=[]

String str = "${uname} : ${password}"

byte [] byt = str.getBytes()

def enc = byt.encodeBase64().toString()

println "Encoded string: " + byt

byte[] dec = enc.decodeBase64()

def res = new String(dec)

end = res.split(':')

println end

**OUTPUT:**

Encoded string: [100, 101, 118, 111, 112, 115, 32, 58, 32, 71, 114, 111, 111, 118, 121]

[devops , Groovy]

1. *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 Sorting\_a\_list {

String name

int days

def getvalues(String n ,int d)

{

def name = n

def days = d

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

}

static void main(args) {

Sorting\_a\_list First=new Sorting\_a\_list()

Sorting\_a\_list Second=new Sorting\_a\_list()

Sorting\_a\_list Third=new Sorting\_a\_list()

Sorting\_a\_list Fourth=new Sorting\_a\_list()

def common = [First.getvalues('maths', 15), Second.getvalues('science', 12), Third.getvalues('english',14), Fourth.getvalues('social',8)]

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

sort.each {println it}

}

}

**OUTPUT:**

[name:maths, days:15]

[name:english, days:14]

[name:science, days:12]

[name:social, days:8]

1. *Operator overloading*
2. 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.
3. Write a MoneyTest class in Groovy that uses + and - and verifies that they work properly.