1. **Number Data Types**
2. 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.

**ANS** - **package** test.demo

**def** x = 2

println x.getClass()

x = 20

println x.getClass()

x = 200

println x.getClass()

x = 200000000000

println x.getClass()

x = 2000000000000000000000

println x.getClass()

x = 2.0

println x.getClass()

x = 20.0

println x.getClass()

x = 20000000000000000000000.0

println x.getClass()

**RESULT:-**

class java.lang.Integer

class java.lang.Integer

class java.lang.Integer

class java.lang.Long

class java.math.BigInteger

class java.math.BigDecimal

class java.math.BigDecimal

class java.math.BigDecimal

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

**ANS - package** test.demo

**def** x = 1+1.5

println x.getClass()

**RESULT:-**

class java.math.BigDecimal

1. 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?

**ANS** - **package** test.demo

**def** x = 5

**def** y = 2

**def** a = x.div(y) // or x / y

**def** b= x.intdiv(y) //Integer Division

println a

println a.getClass()

println b

println b.getClass()

**RESULT:-**

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.

**ANS - package** test.demo

**class** WrapperClass {

**public** **static** **void** main(String[] args) {

println("Byte.MIN = " + Byte.***MIN\_VALUE***);

println("Byte.MAX = " + Byte.***MAX\_VALUE***);

println("Short.MIN = " + Short.***MIN\_VALUE***);

println("Short.MAX = " + Short.***MAX\_VALUE***);

println("Integer.MIN = " + Integer.***MIN\_VALUE***);

println("Integer.MAX = " + Integer.***MAX\_VALUE***);

println("Long.MIN = " + Long.***MIN\_VALUE***);

println("Long.MAX = " + Long.***MAX\_VALUE***);

println("Float.MIN = " + Float.***MIN\_VALUE***);

println("Float.MAX = " + Float.***MAX\_VALUE***);

println("Double.MIN = " + Double.***MIN\_VALUE***);

println("Double.MAX = " + Double.***MAX\_VALUE***);

}

}

**RESULT:-**

Byte.MIN = -128

Byte.MAX = 127

Short.MIN = -32768

Short.MAX = 32767

Integer.MIN = -2147483648

Integer.MAX = 2147483647

Long.MIN = -9223372036854775808

Long.MAX = 9223372036854775807

Float.MIN = 1.4E-45

Float.MAX = 3.4028235E38

Double.MIN = 4.9E-324

Double.MAX = 1.7976931348623157E308

1. **2s Complement**

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

**ANS** - **package** test.demo

**byte** b = 127

println b.getClass()

add = b+1

println add.getClass()

**RESULT –**

class java.lang.Byte

class java.lang.Integer

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

**ANS** - **package** test.demo

**class** GroovyString {

**public** **static** **void** main(String[] args)

{

String str = "Hello, Groovy!";

**int** count = 0;

**for**(**int** i = 0; i < str.length(); i++)

{

**if**(str.charAt(i) != ' ')

count++;

}

println("Total number of characters in the string: " + count);

}

}

**RESULT :-** Total number of characters in the string: 13

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

**ANS - package** test.demo

**def** name = "Kajal"

println "Hello ${name}"

println "Hello ".concat(name)

**RESULT :-** Hello Kajal

Hello Kajal

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

**ANS – def** word = "racecar"

**def** checkDone;

String reverse="";

**int** length = word.size();

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

reverse = reverse + word.charAt(i);

}

**if**(word.equalsIgnoreCase(reverse)){

checkDone = "The String " +word+ " is a palindrome" ;

}**else**{

checkDone = "The String " +word+ " is not a palindrome";

}

println(checkDone )

**RESULT :-** The String racecar is a palindrome

1. 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.

**ANS - package** test.demo

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

println str.split(" ")

println str.split().length

**RESULT :-** [Hello,, World., How, are, you?]

5

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

**ANS** - **package** test.demo

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

println str.substring(7,12)

**RESULT:-** World

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

**ANS - package** test.demo

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

println str.substring(22,25)

println str.substring(22,25).reverse()

**RESULT:-** you

uoy

1. **Prime Numbers**

Write a method called is Prime 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?

**ANS :- class** Prime {

**public** **static** **void** main(String[] args) {

**int** i;

**int** num = 0;

**int** maxCheck = 100;

**boolean** isPrime = **true**;

String primeNumbersFound = "";

**for** (i = 2; i <= maxCheck; i++) {

isPrime = *CheckPrime*(i);

**if** (isPrime) {

primeNumbersFound = primeNumbersFound + i + " ";

}

}

System.***out***.println("Prime numbers from 1 to " + maxCheck + " are:");

System.***out***.println(primeNumbersFound);

}

**public** **static** **boolean** CheckPrime(**int** numberToCheck) {

**int** remainder;

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

remainder = numberToCheck % i;

**if** (remainder == 0) {

**return** **false**;

}

}

**return** **true**;

}

}

**RESULT :-** Prime numbers from 1 to 100 are:

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

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

public class list{

public static void main(String[] args){

List<String> list=new ArrayList<String>();

list.add("Kiwi");

list.add("Pineapple");

list.add("Mango");

list.add("Apple");

list.add("Banana");

list.add("Grapes");

for(String fruit:list)

System.out.println(fruit);

Collections.sort(list);

System.out.println("List after the use of" +

" Collection.sort() :\n" + list);

Collections.sort(list, new Comparator<String>() {

public int compare(String o1, String o2) {

if(o1.length()>o2.length()){

return 1;

}else{

return o1.compareTo(o2);

}

}

});

System.out.println(list);

}

}

**RESULT** :- Kiwi

Pineapple

Mango

Apple

Banana

Grapes

List after the use of Collection.sort() :

[Apple, Banana, Grapes, Kiwi, Mango, Pineapple]

[Apple, Banana, Grapes, Kiwi, Mango, Pineapple]

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.

def myList = ["1", "2", "3","4","5"]

println myList

**RESULT :-** [1, 2, 3, 4, 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.

**ANS -** String filepath = "C:/Users/HP/eclipse-workspace/FirstGroovyProject/data1.txt"

File myFile = **new** File(filepath)

println myFile.text

println myFile.size()

**RESULT:-**

Line 1

Line 2

Line 3

22

1. **Closures as a filter**

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

ANS -

**def** mylist = [1,4,-1,-2,0,5,3,-5,6,2]

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

**RESULT :-** [1, 4, 5, 3, 6, 2]

1. **Multi-line strings**

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

Code: public class filter {

static boolean isVowel(char ch)

{

ch = Character.toUpperCase(ch);

return (ch=='A' || ch=='E' || ch=='I' ||

ch=='O' || ch=='U');

}

static int countVowels(String str)

{

int count = 0;

for (int i = 0; i < str.length(); i++)

if (isVowel(str.charAt(i)))

++count;

return count;

}

public static void main(String[] args)

{

String str1 = "abc ude";

String str2= "raksha";

String str3 = "Capgemini";

System.out.println(countVowels(str1));

System.out.println(countVowels(str2));

System.out.println(countVowels(str3));

}

}

**RESULT** :- 3

2

4

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.

Code: import java.lang.Math;

class Gfg1 {

// driver code

public static void main(String[] args)

{

int l = 1;

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

l = 2;

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

l = 3;

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

l = 4;

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

l = 5;

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

l = 6;

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

l = 7;

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

l = 8;

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

l = 9;

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

l = 10;

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

l = 11;

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

l = 12;

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

l = 13;

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

l = 14;

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

l = 15;

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

}

}

**o/p**: Binary is 1

Binary is 10

Binary is 11

Binary is 100

Binary is 101

Binary is 110

Binary is 111

Binary is 1000

Binary is 1001

Binary is 1010

Binary is 1011

Binary is 1100

Binary is 1101

Binary is 1110

Binary is 1111

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.

Code: public class sum {

public static void main(String[] args) {

int [] arr = new int [] {1, 2, 3, 4, 5};

int sum = 0;

for (int i = 0; i < arr.length; i++) {

sum = sum + arr[i];

}

System.out.println("Sum of all the elements of an array: " + sum);

}

String username = "rakshakakde"

String password = "kakde098"

def man = username+":"+password

println man

String encoded = man.bytes.encodeBase64().toString()

println "after encoding $encoded"

byte[] decoded = encoded.decodeBase64()

println new String (decoded)

String s = new String (decoded)

String[] temp;

String delimeter = ":";

temp = s.split(delimeter);

String username1 = temp[0];

String password2 = temp[1]

println "original username: " +username1

println "original password: " +password2

}

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.

Code: class course {

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) {

course First=new course()

course Second=new course()

course Third=new course()

course Fourth=new course()

def common = [First.getvalues('java', 30), Second.getvalues('python', 30), Third.getvalues('groovy', 20), Fourth.getvalues('unix', 35)]

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

sort. each {println it}

}

}

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.