**GROOVY**

1. *Number Data Types*
   1. What data type is the number 2? How about 20? 200? Keep adding zeros and watch the data type change until it reaches Big Integer. Then do the same for 2.0.

🡪 **def** b = 2

println b.getClass().getName()

Output on console: java.lang.Integer

🡪 **def** b = 20

println b.getClass().getName()

Output on console: java.lang.Integer

🡪**def** b = 2000000000000000

println b.getClass().getName()

Output on console: java.lang.Long

🡪 **def** b = 2000000000000000

println b.getClass().getName()

Output on console: 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?

🡪 **def** **int** x1=1, x2=1.5, sum;

sum = x1 + x2;

System.***out***.println("Sum is "+sum);

println sum.getClass().getName()

Output on console: Sum is 2

java.lang.Integer

* 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?
     + **package** com.test.demo

**class** myfirstgroovyclass {

**static** **void** main(args) {

**def** a = 5

**def** b = 2

**def** result = 5.div(2)

**def** resultinte = 5.intdiv(2)

System.***out***.println(result)

System.***out***.println(resultinte)

println result.getClass().getName()

}

}

Output on console: 2.5

2

java.math.BigDecimal

1. *Wrapper Classes*

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

🡪 **package** com.test.demo

**class** myfirstgroovyclass {

**static** **void** main(args) {

//byte

println ("byte")

println Byte.***MIN\_VALUE***

println Byte.***MAX\_VALUE***

//short

println ("short")

println Short.***MIN\_VALUE***

println Short.***MAX\_VALUE***

//int

println ("int")

println Integer.***MIN\_VALUE***

println Integer.***MAX\_VALUE***

//LONG

println("long")

println Long.***MIN\_VALUE***

println Long.***MAX\_VALUE***

// float

println("Float")

println Float.***MIN\_VALUE***

println Float.***MAX\_VALUE***

// double

println("double")

println Double.***MIN\_VALUE***

println Double.***MAX\_VALUE***

}

}

Output on console: byte

-128

127

short

-32768

32767

int

-2147483648

2147483647

long

-9223372036854775808

9223372036854775807

Float

1.4E-45

3.4028235E38

double

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?

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

🡪 **package** com.test.demo

**class** myfirstgroovyclass {

**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++;

}

System.***out***.println("Lenght of String is: " + count);

}

}

Output on console: Lenght of String is: 13

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

🡪 **package** com.test.demo

**class** myfirstgroovyclass {

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

String str1 = "Kalyani";

String str2 = "Hello";

println(str1 + str2);

}

}

Output on console: KalyaniHello

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

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.

🡪

**package** com.test.demo

**class** myfirstgroovyclass {

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

String a = "Hello, World. How are you?";

String[] str;

str = a.split();

**for**( String values : str )

println(values);

String string = "Hello, World. How are you?";

**int** count = 0;

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

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

count++;

}

System.***out***.println("Total number of word in string is: " +count)

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

println b.tokenize()

}

}

Output on console: Hello,

World.

How

are

you?

Total number of word in string is: 22

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

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

🡪 **package** com.test.demo

**class** myfirstgroovyclass {

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

String a = "Hello, World. How are you?";

println(a.substring(7,12));

}

}

Output on console: World

* 1. Use array notation to print the last word, but reversed.
     + **package** com.test.demo

**class** myfirstgroovyclass {

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

String a = "Hello, World. How are you?";

String lastWord = a.substring(a.lastIndexOf(" ")+1);

println(lastWord.reverse());

}

}

Output on console: ?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?

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

* + - **package** Groovy

**class** sample {

**static** **void** main(args) {

**def** str = ["India", "Australia", "America", "Germany","France"]

**def** a = str.sort();

println("Alphabetically Sorted " +a)

str = str.sort{it.size() }

println("Sorted by size" +str)

str = str.sort{-it.size() }

println("Sorted by Decending order" +str)

}

}

Output on console: Alphabetically Sorted [America, Australia, France, Germany, India]

Sorted by size[India, France, America, Germany, Australia]

Sorted by Decending order[Australia, America, Germany, France, India]

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.

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.

1. *Closures as a filter*

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

1. *Multi-line strings*

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

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.

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.
4. *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.

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.