**Collection Exercise -1**

**1. Initialize an empty list and give the output of the following code:**

**Script:-**

List<String> l= []

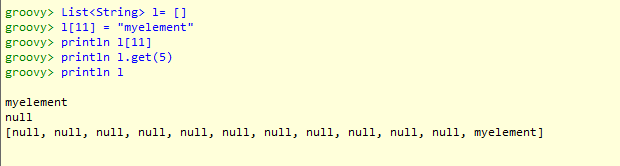
l[11] = "myelement"

println l[11]

println l.get(5)

println list

**OUTPUT:-**

****

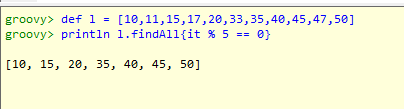
**2. Initialize a list and find all elements which are divisible by 5.**

**Script:-**

def l = [10,11,15,17,20,33,35,40,45,47,50]

println l.findAll{it % 5 == 0}

**OUTPUT:-**

****

**3. Given two lists [11, 12, 13, 14] and [13, 14, 15], how would we obtain the list of items from the first that are not in the second?**

**Script**

List l1=[11,12,13,14]

List l2=[13,14,15]

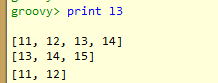
println l1

println l2

List l3=l1-l2

print l3

**output**

****

**4. Find whether two lists have a common element or not.**

**Script**

List list1=[11,12,13,14]

List list2=[13,14,15]

println list1

println list2

List list3 = list1.intersect(list2)

if(list3)

{

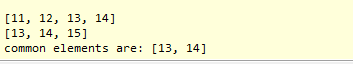
println "common elements are: " + list3

}

else

print "List are Disjoint"

**OUTPUT :-**

****

**5. Remove all records from a list whose index is odd.**

**Script:-**

List list1=(1..10)

println list1

list1.collect{

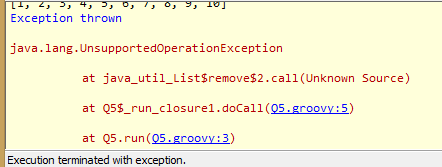
if(it%2!=0)

list1.remove(it)

}

print list1;

**THIS CODE does not work.**

****

**6. Consider the following list:**

**[1, 2, 3, "element1", 0.3, [2, 4, 6], 0..10 ]**

**Print the class name of each element. What's the output of the following statement?**

**list.get(6).get(9)**

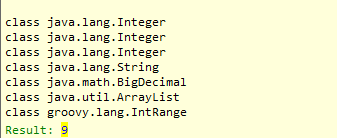
List list1=[1,2,3,"element1",0.3,[2,4,6],0..10]  
list1.each(

{ print it.getClass()

print "\n"

}   
)

list1.get(6).get(9)



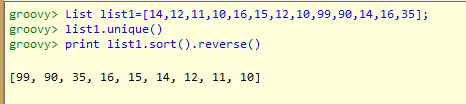
**7. Sort the given list in descending order having distinct elements:**

**[14,12, 11,10, 16, 15,12, 10, 99, 90, 14, 16, 35]**

List list1=[14,12,11,10,16,15,12,10,99,90,14,16,35];

list1.unique()

print list1.sort().reverse()



**8. Consider a class Employee with following details**

**\* Name**

**\* Age**

**\* Salary**

**Create a list consisting of 10 Employee objects.**

**(a). Get a list of employees who earn less than 5000**

class Employee

{

String Name;

byte age;

int salary;

}

Employee e1=new Employee(name:'aa',age:46 as byte,salary:8000)

Employee e2=new Employee(name:'bb',age:37 as byte,salary:7000)

Employee e3=new Employee(name:'cc',age:26 as byte,salary:1800)

Employee e4=new Employee(name:'dd',age:38 as byte,salary:4200)

Employee e5=new Employee(name:'ee',age:24 as byte,salary:6000)

Employee e6=new Employee(name:'ff',age:32 as byte,salary:3000)

Employee e7=new Employee(name:'gg',age:31 as byte,salary:2500)

Employee e8=new Employee(name:'hh',age:35 as byte,salary:8600)

Employee e9=new Employee(name:'ii',age:25 as byte,salary:4800)

Employee e10=new Employee(name:'jj',age:23 as byte,salary:3500)

List l=[e1,e2,e3,e4,e5,e6,e7,e8,e9,e10]

List l2=l.findAll{Employee e->

e.salary<5000

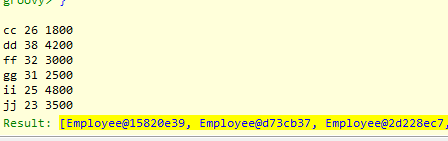
}

l2.each

{

println it.name+" "+it.age+" "+it.salary

}

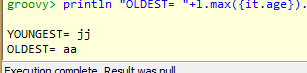
****

**(b). Get the name of the youngest employee and oldest employee**

List l=[e1,e2,e3,e4,e5,e6,e7,e8,e9,e10]

println "YOUNGEST= "+l.min({it.age}).name

println "OLDEST= "+l.max({it.age}).name

****

**(c). Get the employee with maximum salary**

List l=[e1,e2,e3,e4,e5,e6,e7,e8,e9,e10]

println "PERSON WITH HIGHEST SALARY= "+l.max({it.salary}).name

****

**(d). Get the list of names of all the employees**

List l=[e1,e2,e3,e4,e5,e6,e7,e8,e9,e10]

List l1=l.findAll

{

it.name

}

print l1.name



**9. Consider the following piece of code:**

**String s = "this string needs to be split"**

**println s.tokenize(" ")**

**println s.tokenize()**

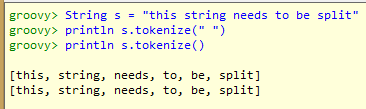
**Compare this with the following code:**

**String s = "this string needs to be split"**

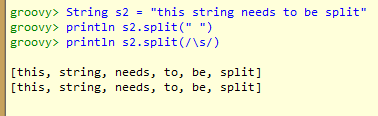
**println s.split(" ")**

**println s.split(/\s/) (Try Same Parameter with tokenize)**

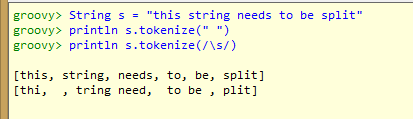
i)



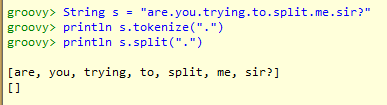
ii)



iii)



iv)



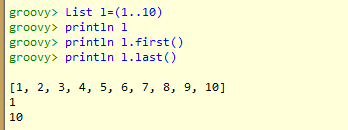
**10. Get first and last element of List.**

List l=(1..10)

println l

println l.first()

println l.last()



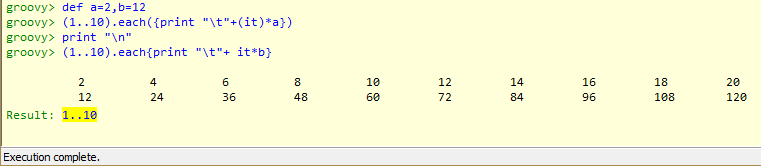
**11. Print the table of a given number : 2 and 12**

def a=2,b=12

(1..10).each({print "\t"+(it)\*a})

print "\n"

(1..10).each{print "\t"+ it\*b}



**12. We have a sorted list of alphabets a-z, print all alphabets appearing after j**

List l=('a'..'z')

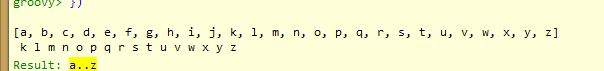
println l

l.each({

if(it>'j')

print " "+it

})



**13. Find the number of occurrences of a character in a string**

String str="this is a simple string to check"

def n='s'

def count=0;

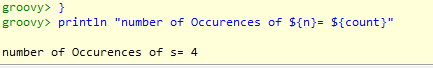
str.each{

if(it==n)

count++

}

println "number of Occurences of ${n}= ${count}"



**14. Write a program that prints the numbers from 1 to 100. But for multiples of three print “Fizz” instead of the number and for the multiples of five print “Buzz”. For numbers which are multiples of both three and five print "FizzBuzz".**

List l=(1..100)

l.each({

if(it%3==0)

print " " + "Fizz"

else if(it%5==0)

print " " + "Buzz"

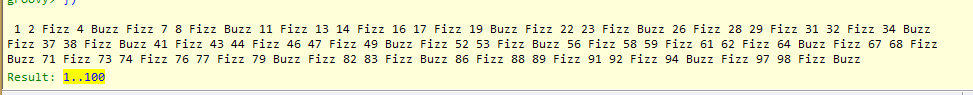
else if(it%15==0)

print " " + "FizzBuzz"

else

print " "+it

})



**15. Consider a class named "Stack" that holds a list of objects and has the following operations associated:**

**1) POP - Pops the last element off the stack**

**2) PUSH - Pushes an element on top of the stack**

**3) TOP - Returns the element at the top of the list**

**Implement the aforesaid class**

class Stack

{

List l=[];

int top=-1

void push(String x)

{

top++;

l.add(x);

println x + " pushed"

}

void pop()

{

if(top>=0)

{

println l[top]+" popped"

l.pop()

top--

}

}

void top()

{

if(top>=0)

{

println l.last()

}

else

println "list is empty"

}

}

Stack obj=new Stack();

obj.push("aaa")

obj.push("bbb")

obj.push("ccc");

obj.pop();

obj.top();

obj.pop();

obj.top();

obj.pop();

obj.top();

