**SOLUTIONS:**

1)

List l=[1,2,2,3,4,4,5]

Set s=l

println s

//I observed that when list is assigned to a set it contains unique value. (duplicate items are removed)

List l=[1,2,2,3,4,4,5]

println l.unique()

//Achieving uniqueness in a list through unique method.

2)

Range r=('a'..'z')

println r.from

println r.to

3)

println "TABLE OF 2"

(1..10).each

{

println "2 \* ${it}=${2\*it}"

}

println "TABLE OF 12"

(1..10).each

{

println "12 \* ${it}=${12\*it}"

}

return

4)

Range r=('a'..'z')

List l=r

pos=l.indexOf('j')

size=l.size()

println l.subList(pos+1,size)

5)

Map m=[deepti:'21',priya:'22',sweta:'23',aditya:'28',rishabh:'27',tarun:'26',sowmya:'19',anusha:'30',duke:'25',ritika:'26']

println m

m.each{

println it

}

6)

Map m1=[deepti:'21',priya:'22',sweta:'23',aditya:'28',rishabh:'27',tarun:'26',sowmya:'19',anusha:'30',duke:'25',ritika:'26']

Map m2=[1:'a']

m2.put("deepti",'22')

m2.put("priya","25")

println m2

Map m3=m1+m2

println m3

**OUTPUT:**

[1:a, deepti:22, priya:25]

[deepti:22, priya:25, sweta:23, aditya:28, rishabh:27, tarun:26, sowmya:19, anusha:30, duke:25, ritika:26, 1:a]

7)

Map map=[:]

println map.class

println map.getClass()

**OUTPUT:**

null

class java.util.LinkedHashMap //In the first case class named key is searched in a map. Since map doesnot contain such key so it returns null. In the second case class of the map is printed which is LinkedHashMap in java.util package.

8)

Map m=['1':2,'2':3,'3':4,'2':5]

println m['2']

**OUTPUT:**

5

It is not a valid construction because map cares about unique identifiers and each key can map atmost one value. Hence value for key '2' is taken as 5 (not as 3)

9)

Map m=['1':2,'2':3,'3':4,'2':5]

println m.containsKey('3')

OUTPUT: true

10)

Map m=['Computing': ['Computing':600, 'Information Systems':300], 'Engineering' : ['Civil':200 , 'Mechanical' :100],'Management':['Management':800]]

println "Departments are:" + m.keySet()

println "Programs delivered by computing department are:" + m.get('Computing')

println "Number of Students enrolled in civil engineering departments:" + m.Engineering.get('Civil')

**OUTPUT:**

Departments are:[Computing, Engineering, Management]

Programs delivered by computing department are:[Computing:600, Information Systems:300]

Students enrolled in civil engineering departments:200

11)

public class Employee

{

String name

Integer age

String dept\_name

Integer emp\_num

Integer salary

public static void main(String[] args)

{

Employee e1=new Employee(name:"deepti ",age:24,dept\_name:"computing",emp\_num:2,salary:10000)

Employee e2=new Employee(name:"priya",age:18,dept\_name:"engineering",emp\_num:3,salary:15000)

Employee e3=new Employee(name:"sweta",age:29,dept\_name:"engineering",emp\_num:4,salary:20000)

Employee e4=new Employee(name:"kanika",age:30,dept\_name:"management",emp\_num:5,salary:21000)

Employee e5=new Employee(name:"ashima",age:23,dept\_name:"computing",emp\_num:1,salary:2000)

Employee e6=new Employee(name:"anusha",age:40,dept\_name:"management",emp\_num:6,salary:50000)

}

}

(a) List l=[e1,e2,e3,e4,e5,e6]

Map m1=l.groupBy{

switch(it.salary){

case (0..5000):return "0-5000"

case (5001..10000):return "5001-10000"

case (10001..20000): return "10001-20000"

default: return ">20000"

}

}

m1.each{

println "Salary Category is $it.key and Employee in these category are:"

println it.value\*.name

}

OUTPUT:

Salary Category is 5001-10000 and Employee in these category are:

[deepti ]

Salary Category is 10001-20000 and Employee in these category are:

[priya, sweta]

Salary Category is >20000 and Employee in these category are:

[kanika, anusha]

Salary Category is 0-5000 and Employee in these category are:

[ashima]

(b)

Map m2=l.countBy{it.dept\_name}

m2.each{

println "Department Name:$it.key ,Number of Employee: $it.value"

}

OUTPUT:

Department Name:computing ,Number of Employee: 2

Department Name:engineering ,Number of Employee: 2

Department Name:management ,Number of Employee: 2

(c)

println "Age between 18 and 35"

l.findAll{it.age>=18 && it.age<=35}.each{ println "$it.name: $it.age"}

OUTPUT:

Age between 18 and 35

deepti : 24

priya: 18

sweta: 29

kanika: 30

ashima: 23

(d)

Map m3=l.groupBy{it.name[0]}

m3.each{

print "Name Start with:$it.key and number of employee in this group with Age >20 are:"

println it.value.countBy{emptit->

emptit.age>20

}.get(true)

}

OUTPUT:

Name Start with:d and number of employee in this group with Age >20 are:1

Name Start with:p and number of employee in this group with Age >20 are:null

Name Start with:s and number of employee in this group with Age >20 are:1

Name Start with:k and number of employee in this group with Age >20 are:1

Name Start with:a and number of employee in this group with Age >20 are:2

(e)

List l=[e1,e2,e3,e4,e5,e6]

def clo= l.groupBy{it.dept\_name}

clo.each{

println it.key

println it.value\*.name

}

OUTPUT:

computing

[deepti , ashima]

engineering

[priya, sweta]

management

[kanika, anusha]

12)

String s= "http://www.google.com?name=johny&age=20&hobby=cricket";

def base =(s.tokenize("?"))[0]

def param=( s.tokenize("?"))[1].tokenize("&")

Map map=[:]

param.each{

def l=it.split("=")

map[(l[0])]=l[1]

}

println map

OUTPUT:

[name:johny, age:20, hobby:cricket]