Groovy collection lab exercise-1

**Introduction to Groovy**

Q1:- Create a class in Java along with follwing fields. classname: Person fields: name, age, gender, address Access the fields in all known ways: like through getter, by dot operator

Sol:

class Person

{

private String name;

private int age;

private String gen;

private String add;

public String getname()

{

return name;

}

public int getage()

{

return age;

}

public String getgender()

{

return gen;

}

public String getaddress()

{

return add;

}

public static void main(String...a)

{

def obj=new Person();

obj.name="yogesh";

obj.age=23;

obj.gen="male";

obj.add="Delhi";

System.out.println("NAME: "+obj.getname());

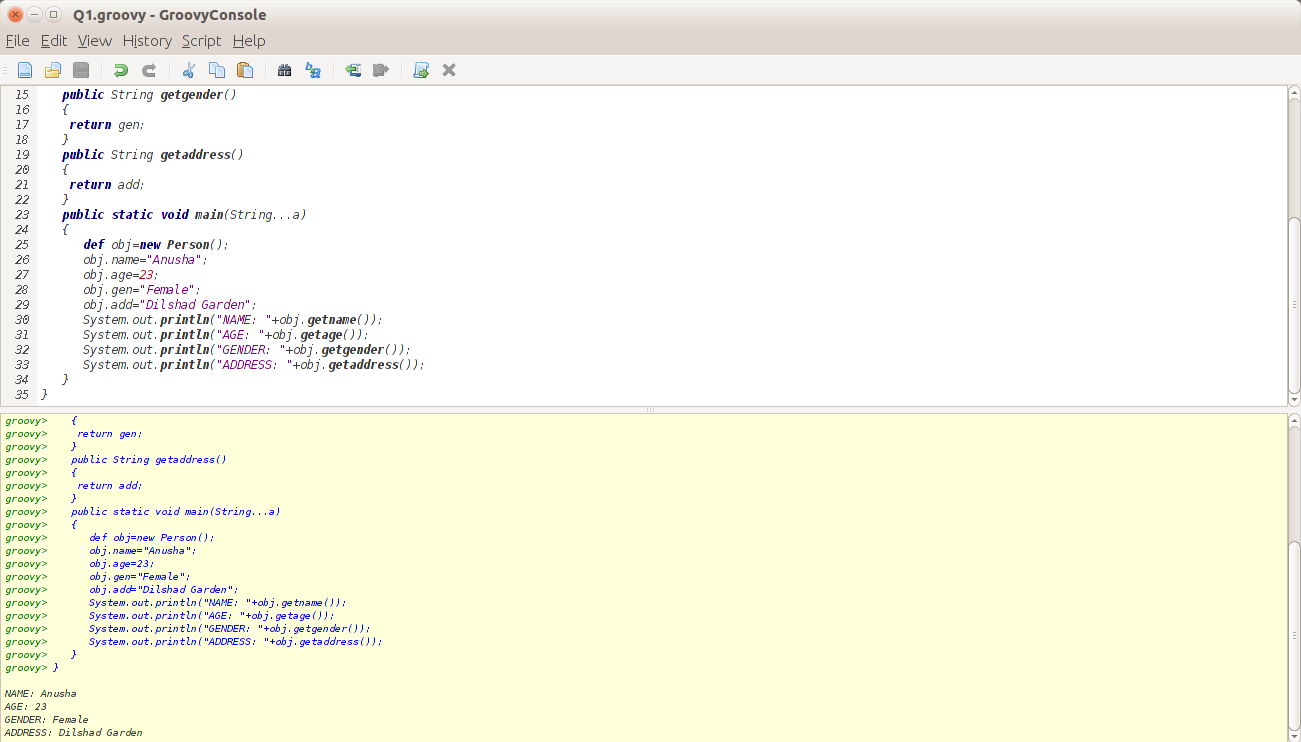
System.out.println("AGE: "+obj.getage());

System.out.println("GENDER: "+obj.getgender());

System.out.println("ADDRESS: "+obj.getaddress());

}

}



2. Print this pattern :

\*

\*\*

\*\*\*\*

\*\*\*\*\*\*\*\*

SoL:

def a;

4.times

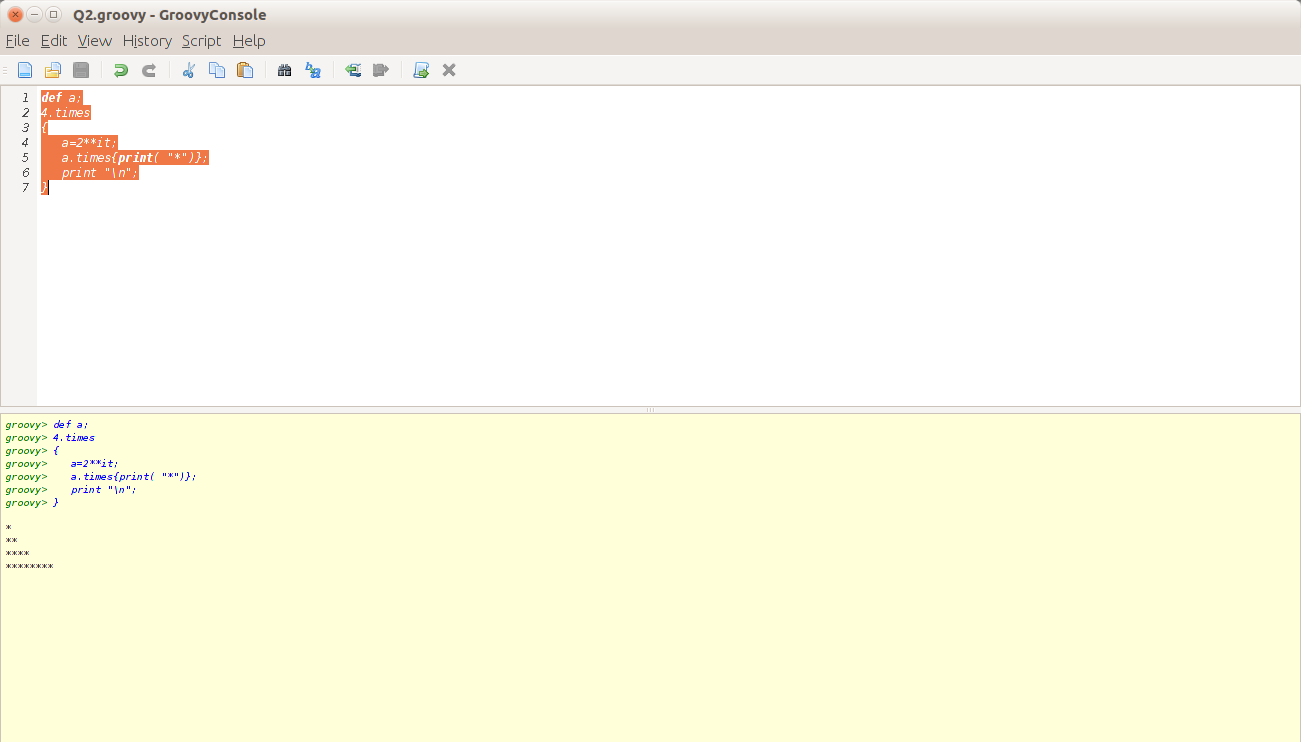
{

a=2\*\*it;

a.times{print( "\*")};

print "\n";

}



3. GString... override the toString() of the Person class to return something like... "Sachin is a man aged 24 who lives at Delhi. He works for Intelligrape with employee id 12 and draws $$$$$$$ lots of money !!!!."

Sol:-

class Ex\_gs

{

String name;

String gen;

int age;

String add;

String comp;

int id;

int sal;

public String toString()

{

print(""+name+" is a "+gen+" aged "+age+\

" who lives at "+add+". He works for "+comp+" employee id "+id+\

" and draws "+sal+" lots of money!!!");

}

public static void main(String...a)

{

def obj=new Ex\_gs();

obj.name="Sachin";

obj.gen="man";

obj.age=24;

obj.add="Delhi";

obj.comp="Intelligrape";

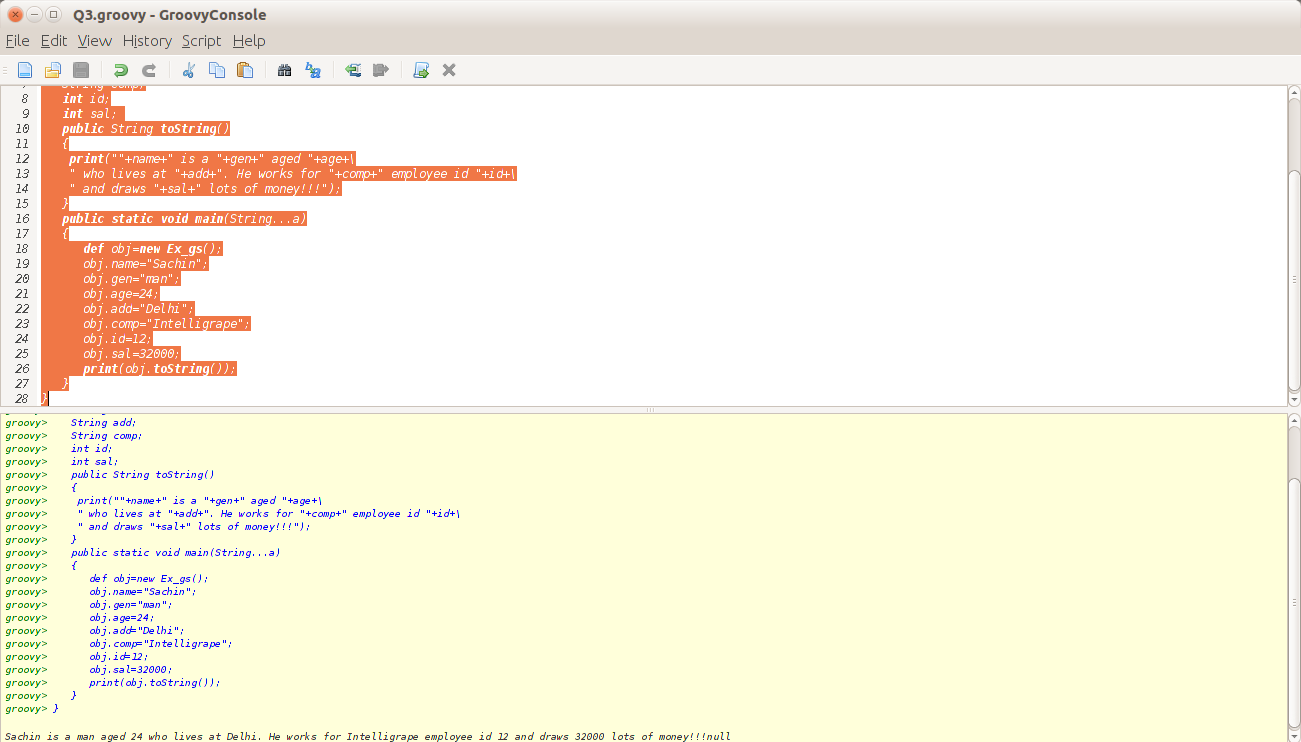
obj.id=12;

obj.sal=32000;

print(obj.toString());

}

}



4. Groovy Truth: if('test') { printlnn "test evaluated to true inside if" }

try replacing test with various objects and observe its behaviour.

a) "Test"

b)'null'

c) null

d) 100

e) 0

f) empty list

g) list with all vaues as null List list = new ArrayList()

Sol:-

if('test')

println "test evaluated to true inside if"

if("Test")

println "test evaluated to true inside if"

if('null')

println "test evaluated to true inside if"

if(null)

println "test evaluated to true inside if"

else

println "test evaluated to false inside if"

if(100)

println "test evaluated to true inside if"

if(0)

println "test evaluated to true inside if"

else

println "test evaluated to false inside if"

List l=[]

if(l)

println "test evaluated to true inside if"

else

println "test evaluated to false inside if"

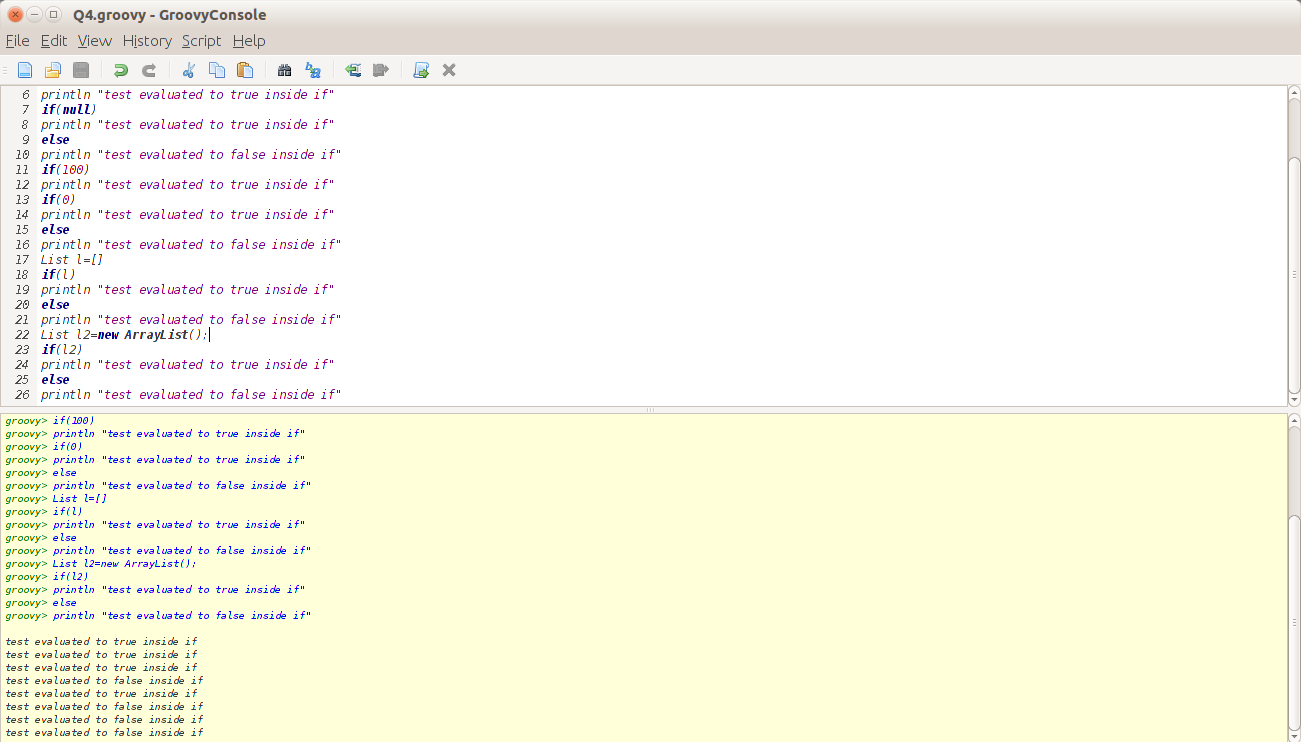
List l2=new ArrayList();

if(l2)

println "test evaluated to true inside if"

else

println "test evaluated to false inside if"



5. Print multiple of 3 upto 10 terms in at least three different ways using groovy special methods

Sol:-

10.times

{

print(3\*(it+1));

print("\t");

}

print "\n"

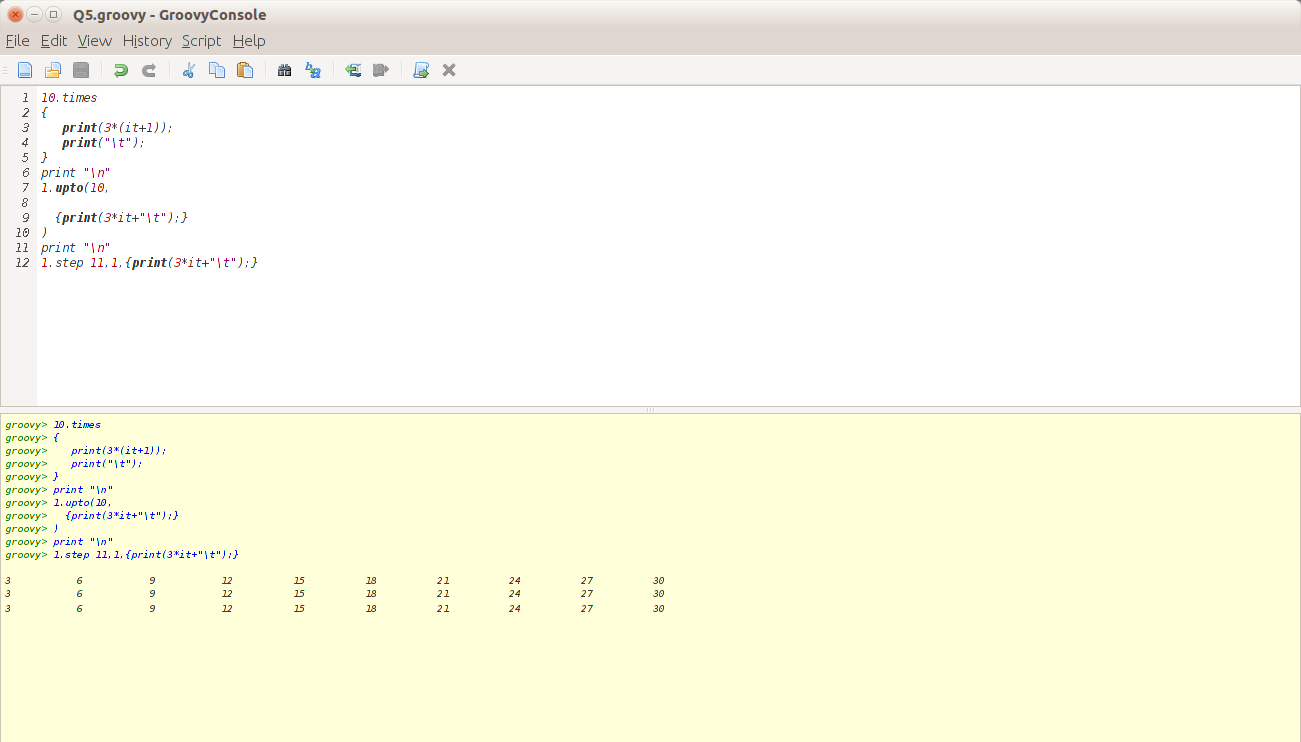
1.upto(10,

{print(3\*it+"\t");}

)

print "\n"

1.step 11,1,{print(3\*it+"\t");}



6. Write a closure which checks if a value is odd/even, Use this closure with list of values to identify which values are odd and which are even.

Sol:-

List l=(1..10)

print l;

print "\n"

println l.groupBy{

if(it%2==0)

{

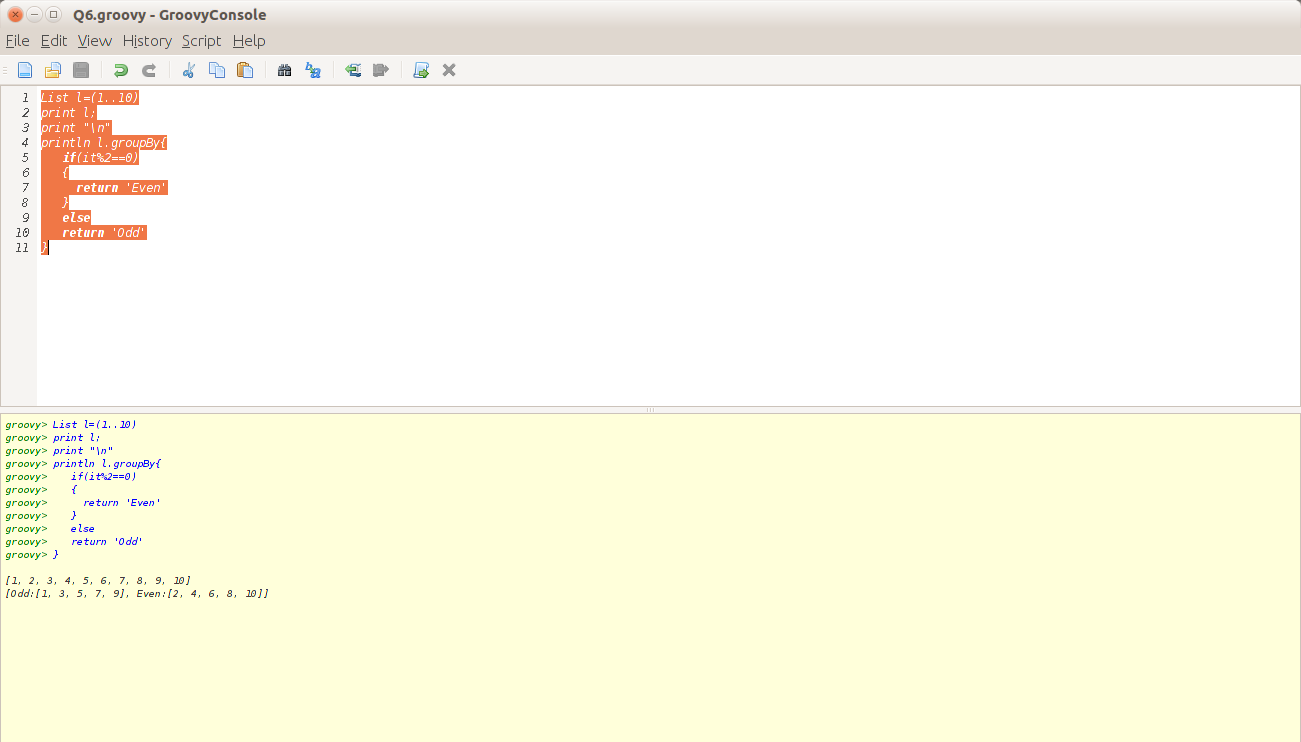
return 'Even'

}

else

return 'Odd'

}



7. Combine content of all the files in a specific directory to another new file

dirName ="file"

def nfile=new File("file1.txt")

new File(dirName).eachFile()

{

file-> file.eachLine{line->nfile<<line<<"\n";}

}

8. Create a file which contains all the odd numbered lines of a given file. Each line should be numbered at the beginning of line viz : 1, 3, 5.....

Sol:-

def nfile=new File('file1.txt')

def file=new File('file1.txt')

int var=0;

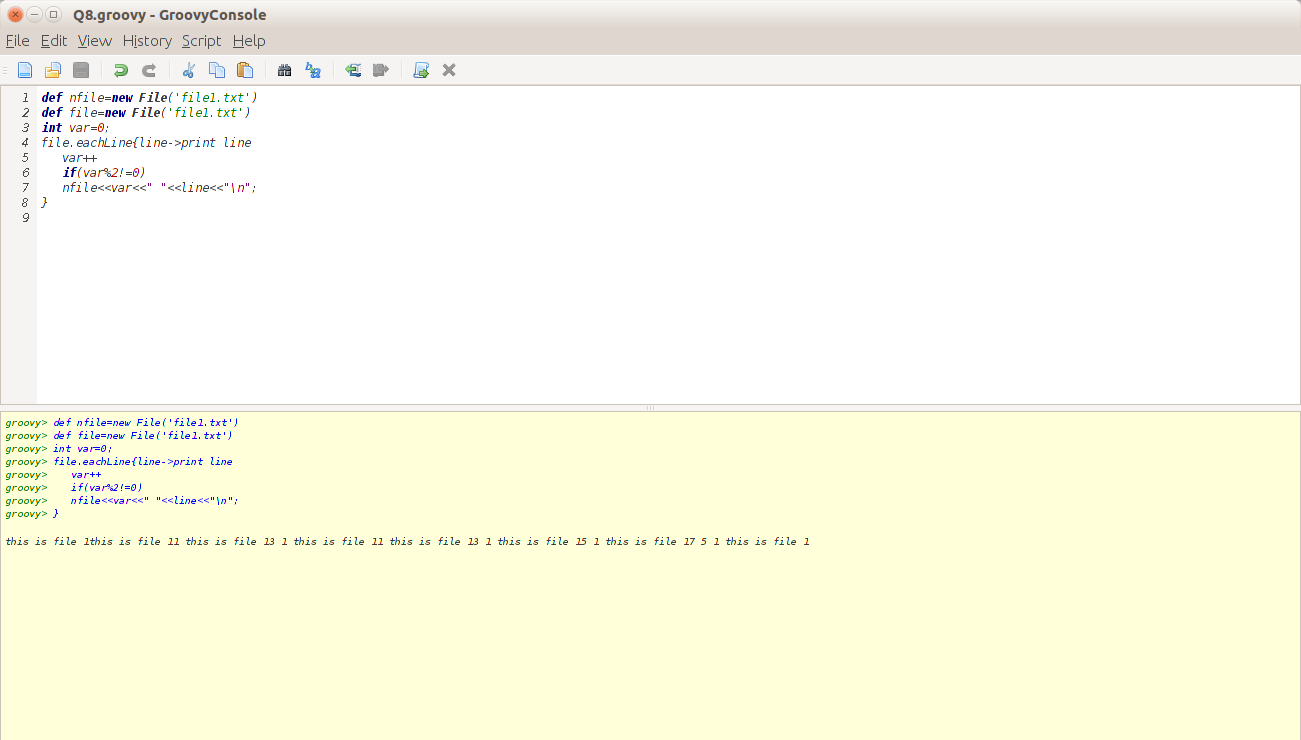
file.eachLine{line->print line

var++

if(var%2!=0)

nfile<<var<<" "<<line<<"\n";

}



9. Write a method which removes all the white spaces in a file and writes the output to another file. Suppose white space characters are Space, Tab and Enter

Sol:-

def file=new File('file1.txt')

def nfile=new File('new2.txt')

def x

file.eachLine{line->print line

x=line.replaceAll("\\s","")

nfile<<x;

}

10. Make copy of an image type.

Sol:-

File a=new File('1.png');

File b=new File('new.png');

**Collections - I**

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

l[11] = "myelement"

println l[11]

println l.get(5)

println l

Sol:-

List l=[]

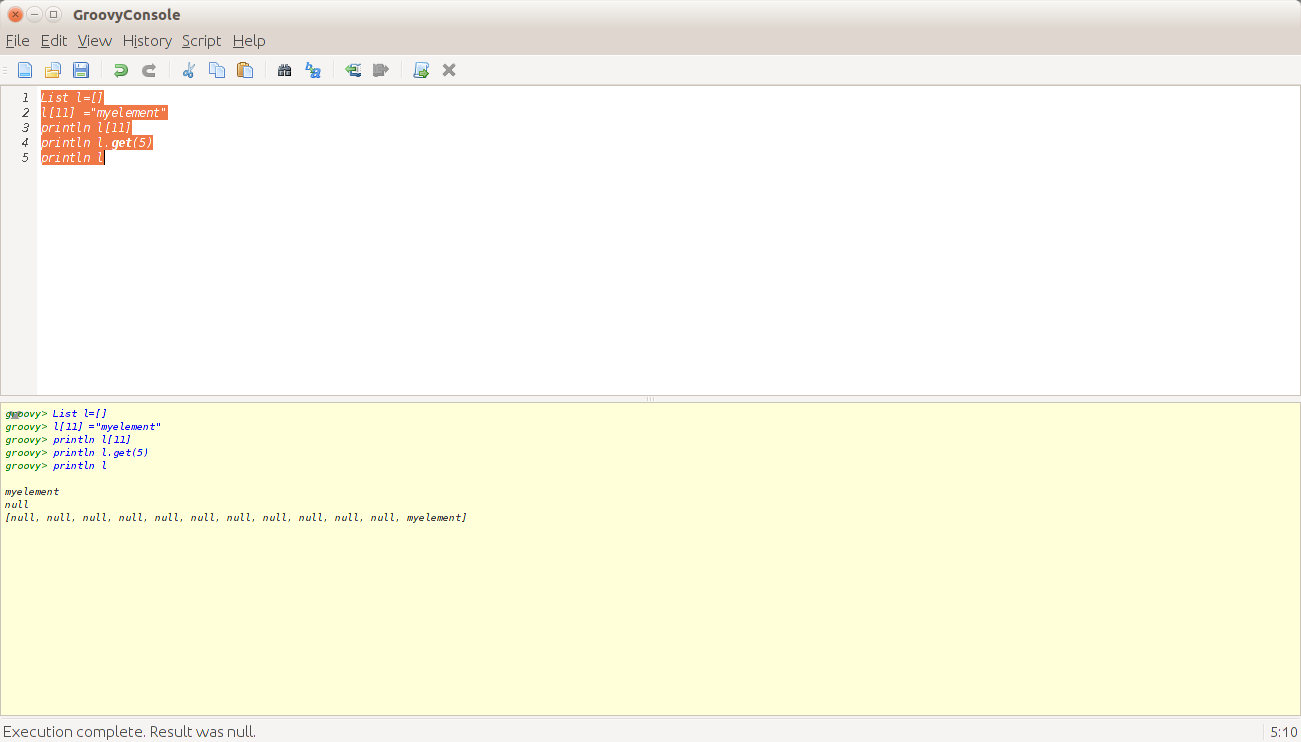
l[11] ="myelement"

println l[11]

println l.get(5)

println l

Output:-



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

Sol:-

List l=(1..20)

l.each

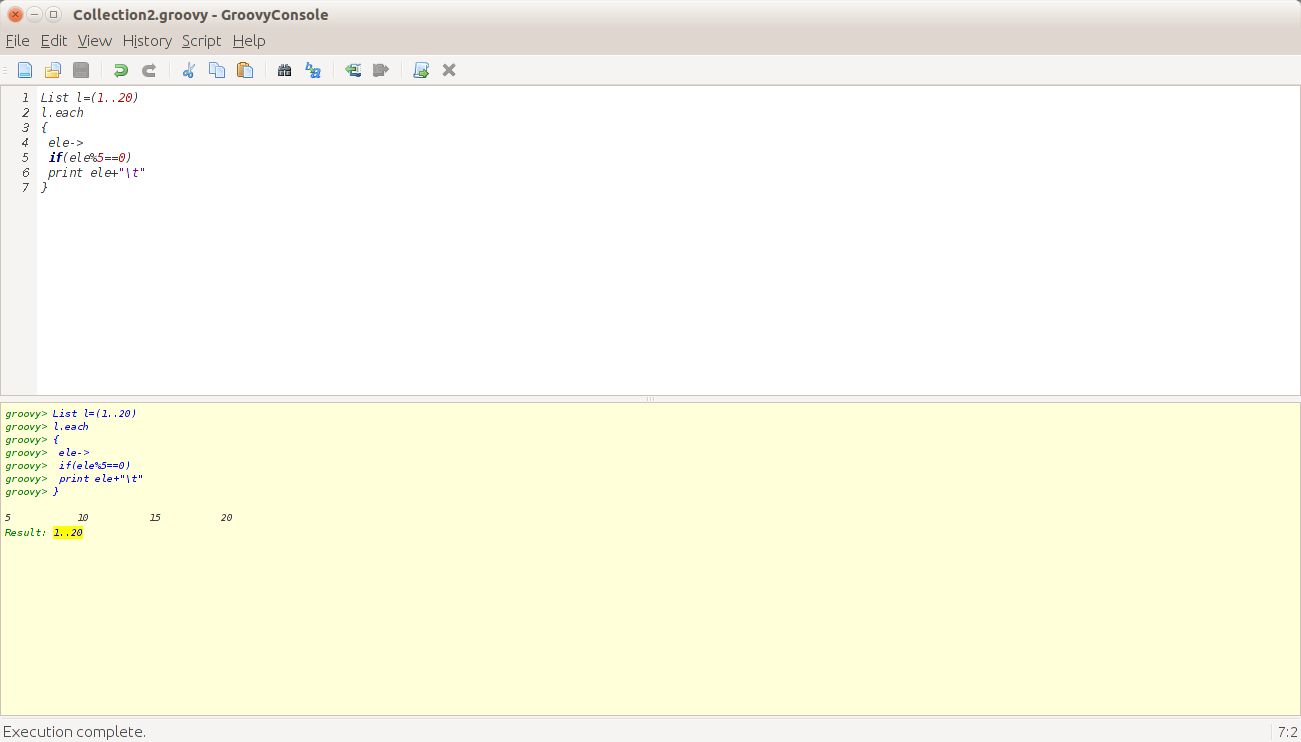
{

ele->

if(ele%5==0)

print ele+"\t"

}



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?

Sol:-

List l1=(11..14)

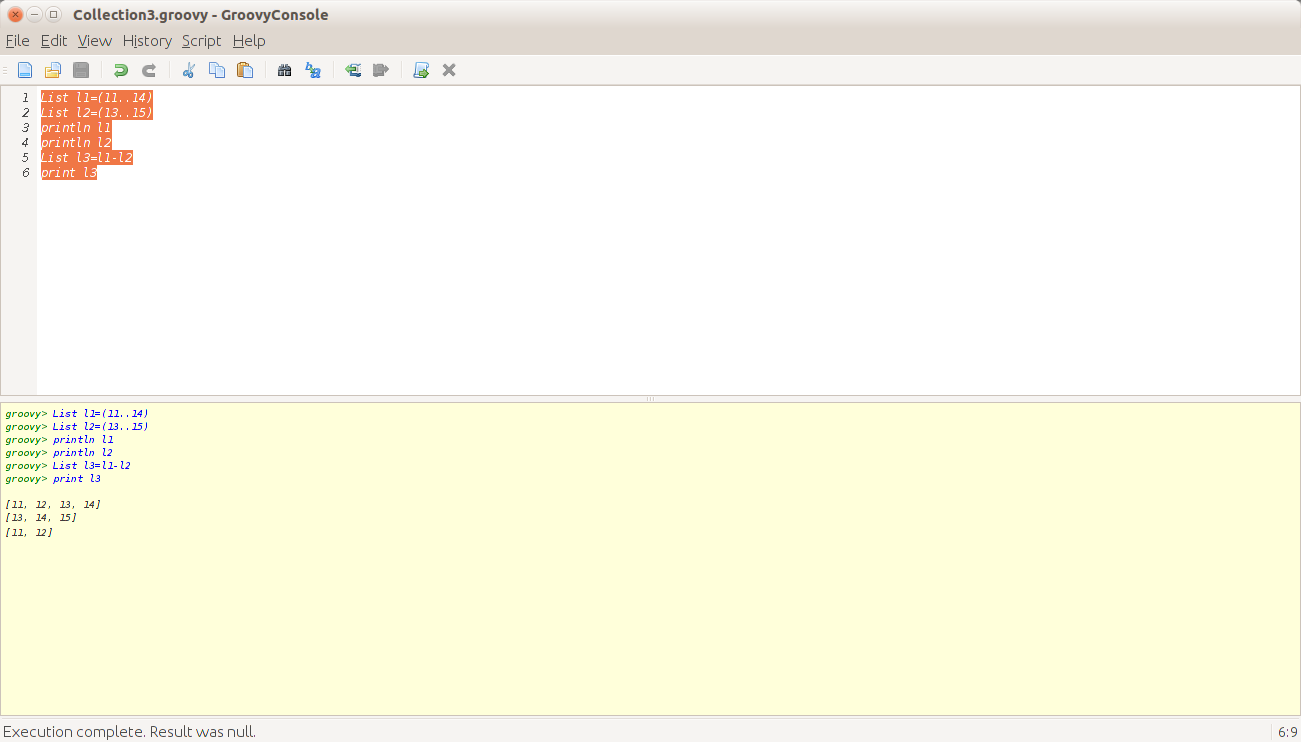
List l2=(13..15)

println l1

println l2

List l3=l1-l2

print l3



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

Sol:-

List l1=(11..14)

List l2=(13..15)

println l1

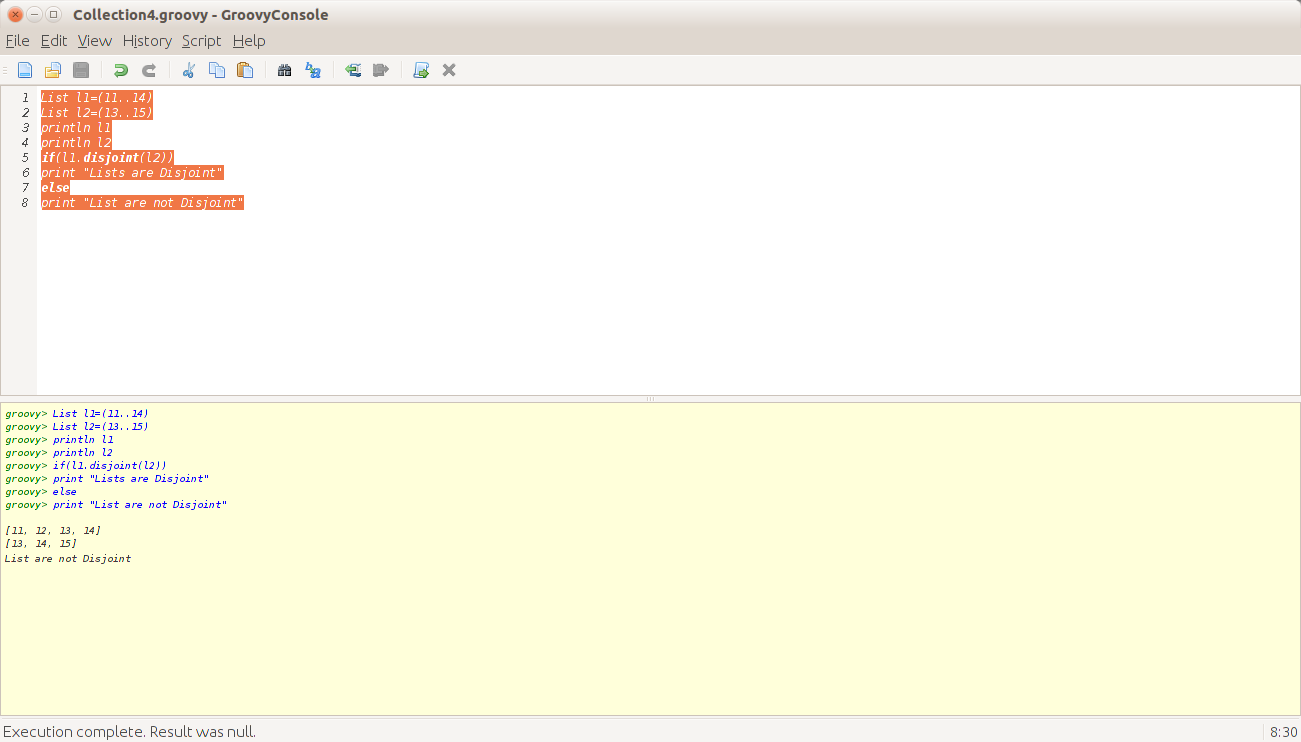
println l2

if(l1.disjoint(l2))

print "Lists are Disjoint"

else

print "List are not Disjoint"



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

Sol:-

List l1=(1..10)

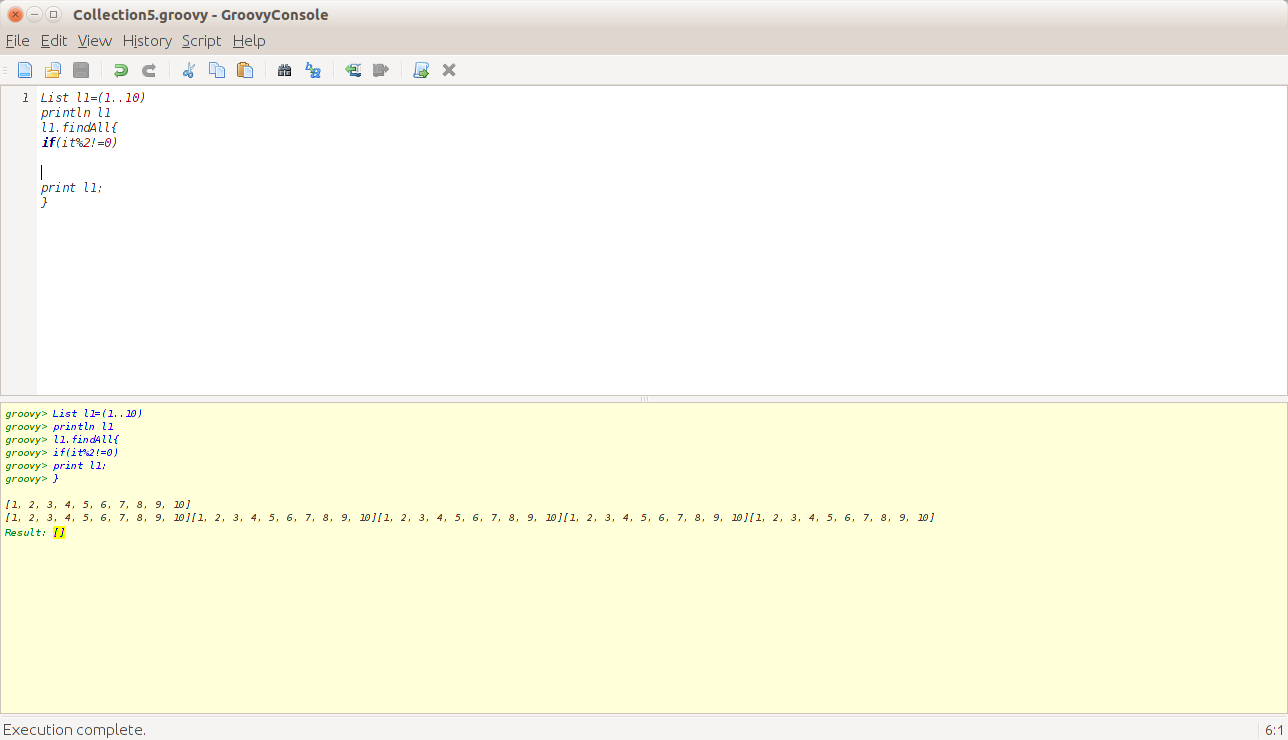
println l1

l1.findAll{

if(it%2!=0)

print l1;

}



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)

Sol:-

List l=[1,2,3,"element1",0.3,[2,4,6],0..10];

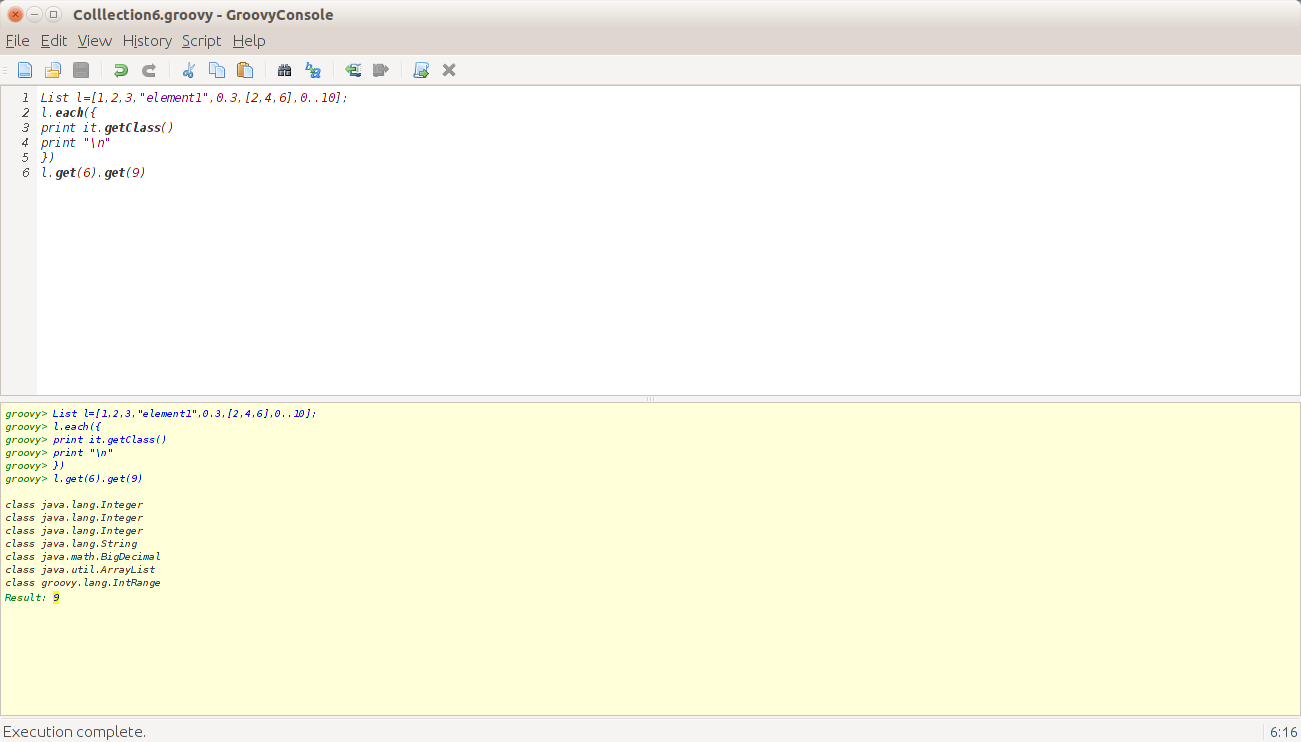
l.each({

print it.getClass()

print "\n"

})

l.get(6).get(9)



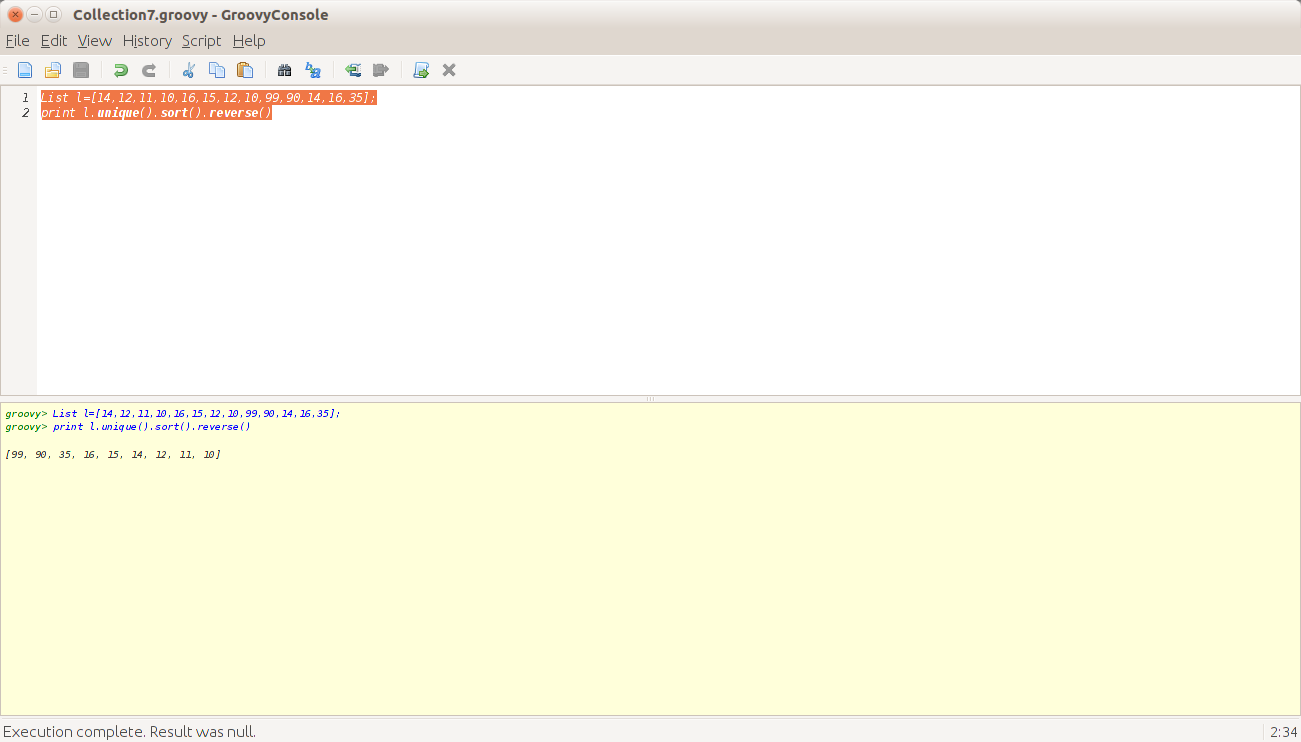
7. Sort the given list in descending order having distinct elements:

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

Sol:-

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

print l.unique().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

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

(c). Get the employee with maximum salary

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

Sol:-

class Employee

{

String Name;

byte age;

int salary;

}

Object[] obj=

[

new Employee(name:'Anusha',age:23 as byte,salary:350000),

new Employee(name:'Nimisha',age:19 as byte,salary:430000),

new Employee(name:'Sudhakaran',age:53 as byte,salary:500000),

new Employee(name:'Geetha',age:45 as byte,salary:300000),

new Employee(name:'Akanksha',age:23 as byte,salary:350000),

new Employee(name:'Pooja',age:22 as byte,salary:455000),

new Employee(name:'Mohit',age:24 as byte,salary:500000),

new Employee(name:'Varun',age:19 as byte,salary:55000),

]

8.times

{

print obj[it].Name+"\t"

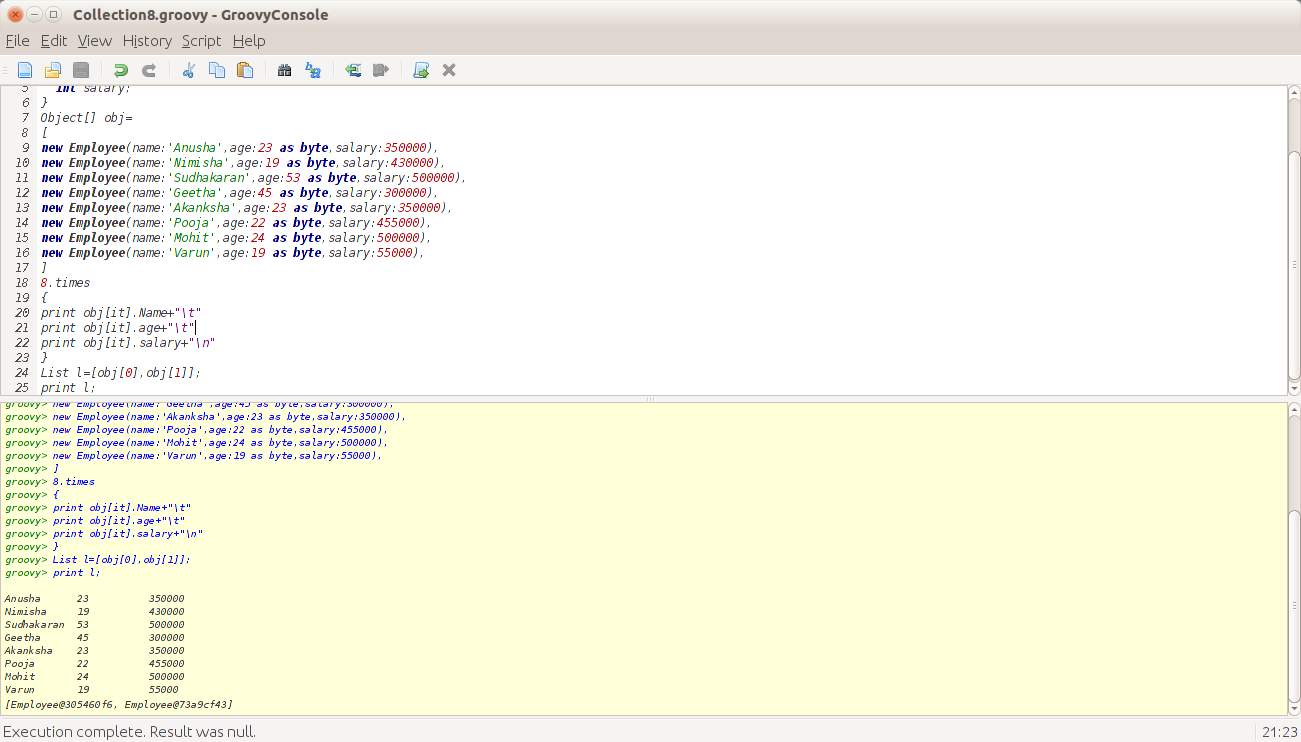
print obj[it].age+"\t"

print obj[it].salary+"\n"

}

List l=[obj[0],obj[1]];

print l;



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)

Also try the following exercise:

String s = "[are.you.trying.to.split.me](http://are.you.trying.to.split.me/).si

s.tokenize(".")

s.split(".")

Sol:-

String s = "this string needs to be split"

println s.tokenize(" ")

println s.tokenize()

String s2 = "this string needs to be split"

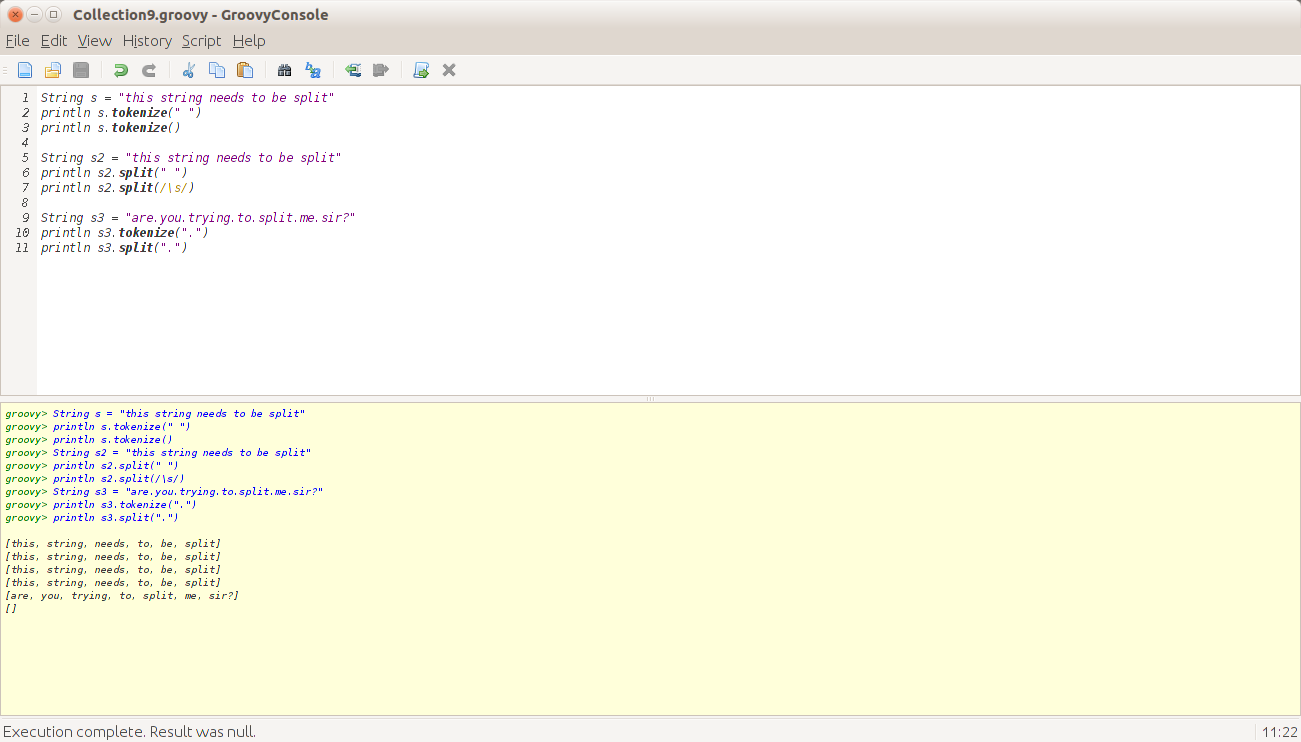
println s2.split(" ")

println s2.split(/\s/)

String s3 = "are.you.trying.to.split.me.sir?"

println s3.tokenize(".")

println s3.split(".")



10. Get first and last element of List.

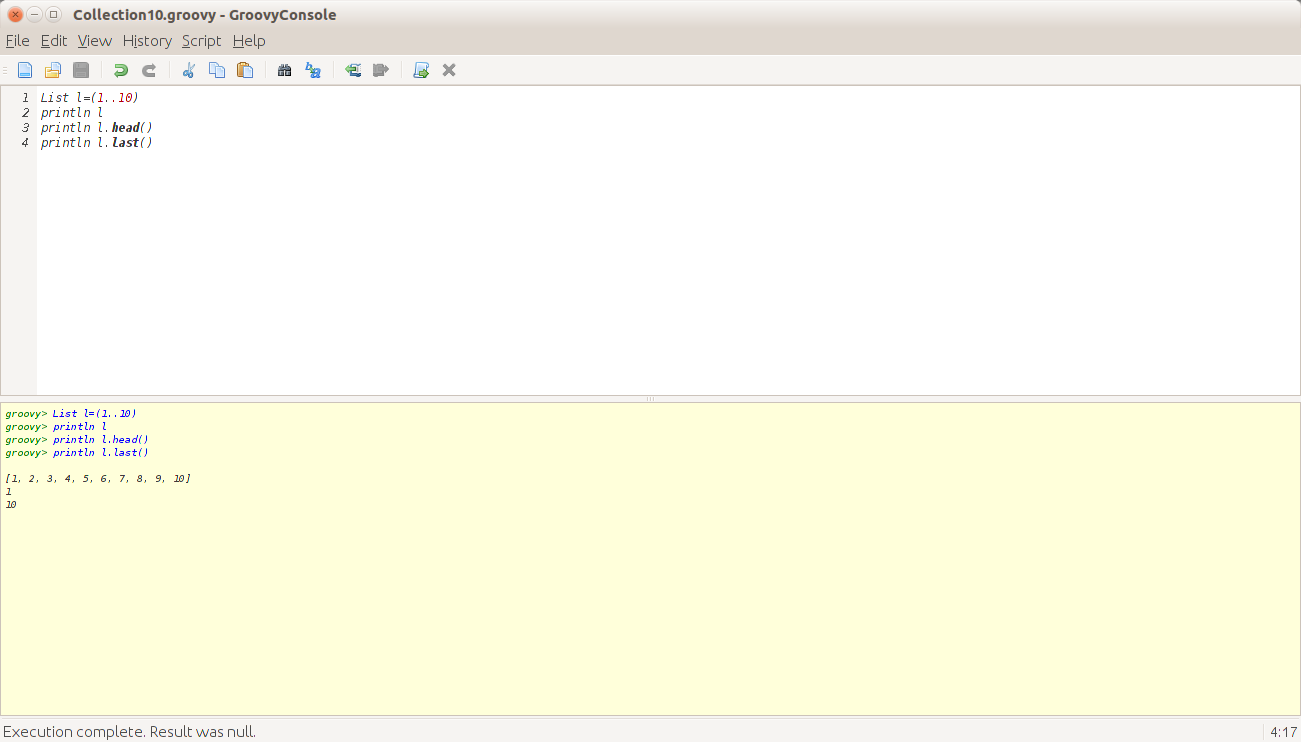
Sol:-

List l=(1..10)

println l

println l.head()

println l.last()



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

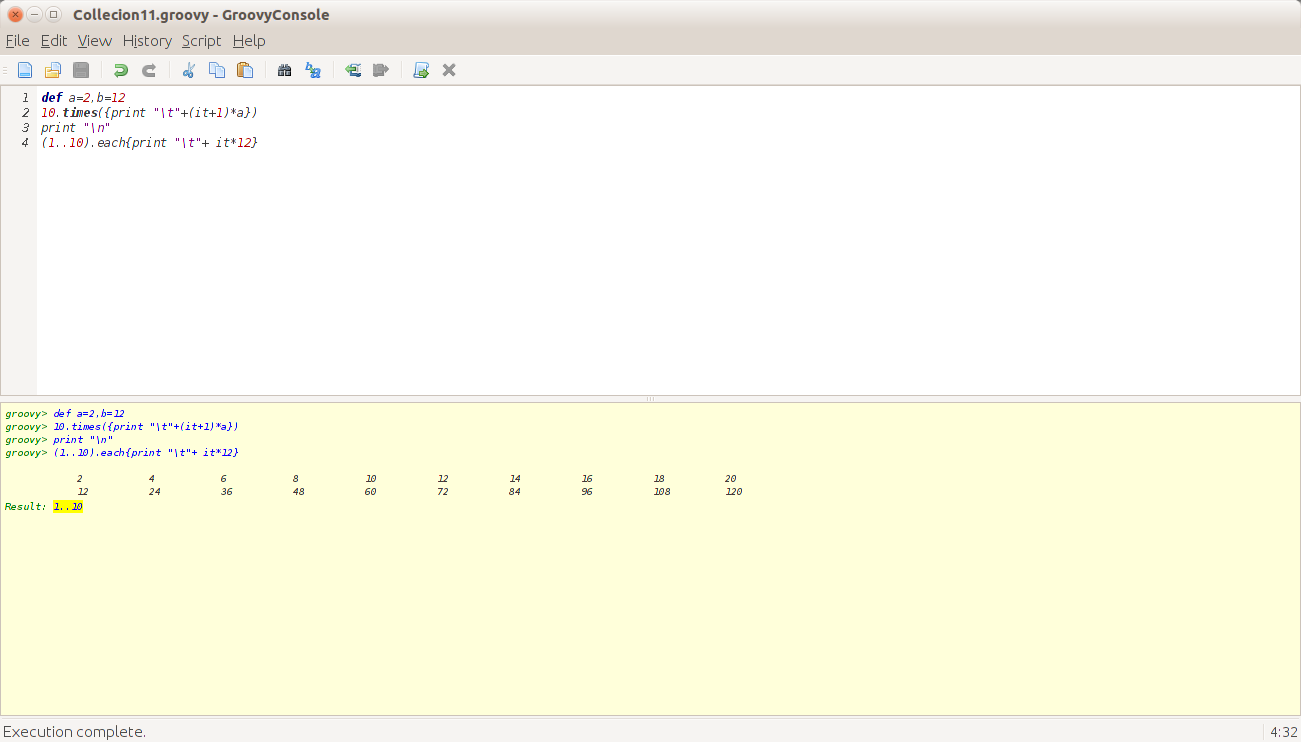
Sol:-

def a=2,b=12

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

print "\n"

(1..10).each{print "\t"+ it\*12}



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

Sol:-

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

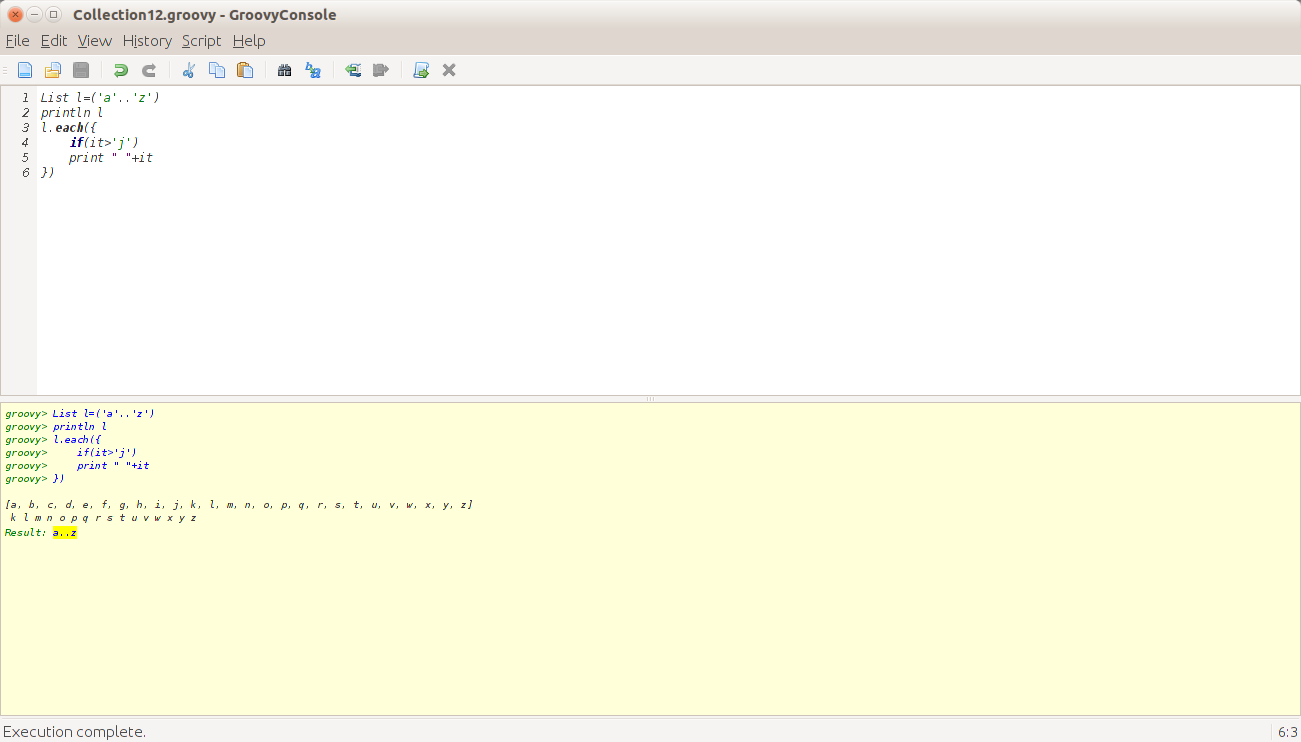
println l

l.each({

if(it>'j')

print " "+it

})



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

Sol:-

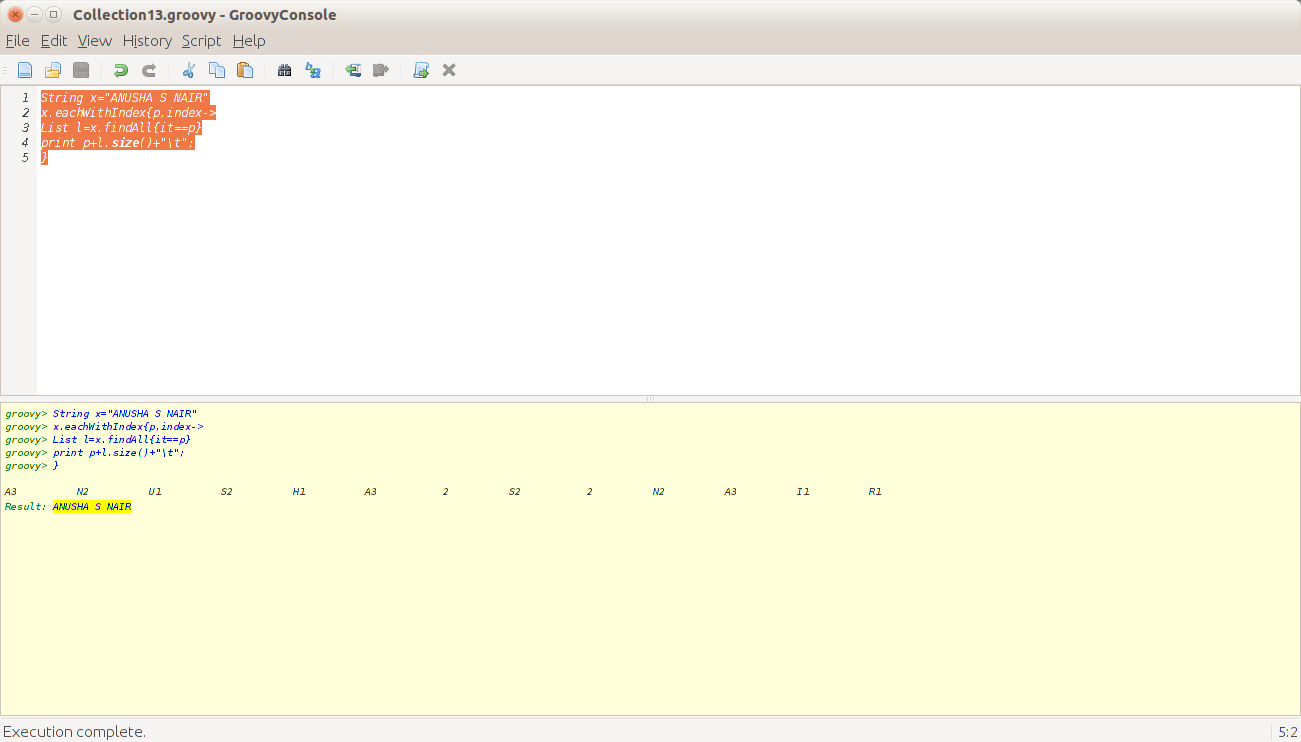
String x="ANUSHA S NAIR"

x.eachWithIndex{p,index->

List l=x.findAll{it==p}

print p+l.size()+"\t";

}



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".

 Sol:-

List l=(1..100)

println l

l.each({

if(it%15==0)

print " "+"FizzBuzz"

else if(it%3==0)

print " "+"Fizz"

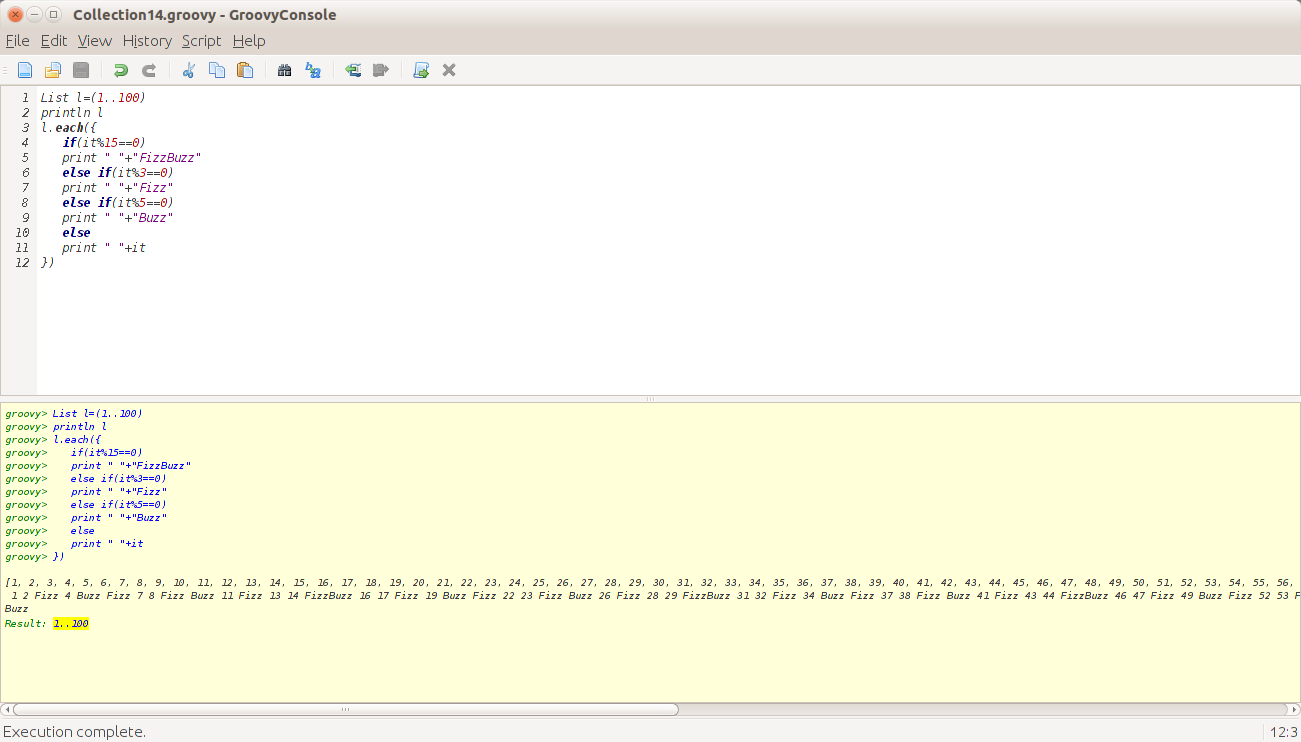
else if(it%5==0)

print " "+"Buzz"

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

Sol:-

class Stack

{

Object[] o=[1,2,3,4,5];

void push(Object x)

{

}

void pop()

{

println o.last();

o.dropRight(1);

}

Object top()

{

return o.last();

}

}

def obj=new Stack();

println obj.o

println obj.top()

obj.pop()

println obj.o

