Rishabh Rajpurohit
Groovy Assignment 1 Solutions

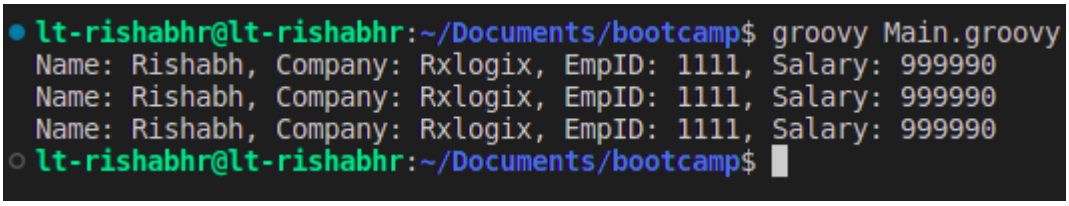
A1.

```
//Person.java
public class Person{
    private String name;
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    private Integer age;
    public Integer getAge() {
        return age;
    }
    public void setAge(Integer age) {
        this.age = age;
    }
    private String gender;
    public String getGender() {
        return gender;
    }
    public void setGender(String gender) {
        this.gender = gender;
    }
    private String address;
    public String getAddress() {
        return address;
    }
    public void setAddress(String address) {
        this.address = address;
    }
    Person(){};
    Person(String name, Integer age, String gender, String address){
        this.name = name;
        this.age = age;
        this.gender = gender;
        this.address = address;
    }
    static public void main(String[] args){
        System.out.println("Class Person created successfully.");
    }
}

//Main.groovy
Person p = new Person("Rishabh",22,"Male","Delhi");
println "Name: ${p.name}, Age: ${p.age}Yrs, Gender: ${p.gender}, Address: $
{p.address}"
println "Name: ${p.getName()}, Age: ${p.getAge()}Yrs, Gender: ${p.getGender()},
Address: ${p.getAddress()}"
```

```
lt-rishabhr@lt-rishabhr:~/Documents/bootcamp$ groovy Main.groovy
Name: Rishabh, Age: 22Yrs, Gender: Male, Address: Delhi
Name: Rishabh, Age: 22Yrs, Gender: Male, Address: Delhi
lt-rishabhr@lt-rishabhr:~/Documents/bootcamp$
```

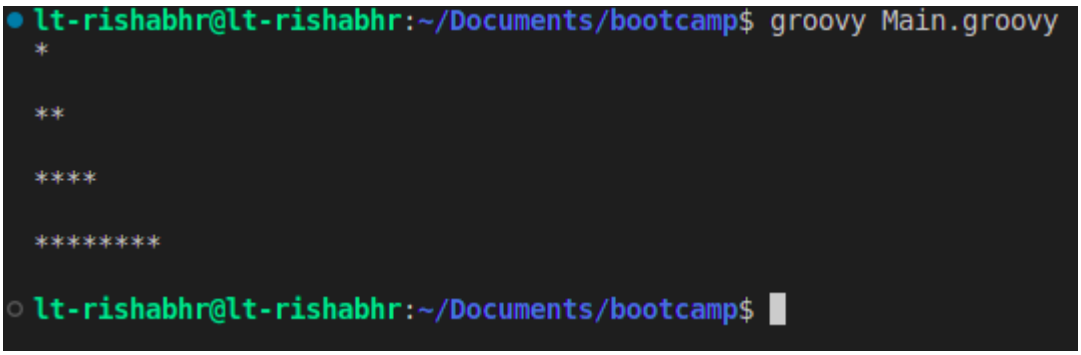
A2.

```
class Employee extends Person{
    Integer empid
    String company
    Integer salary
    Employee(){}
    Employee(String name, Integer age, String gender, String address, Integer
empid, String company, Integer salary){
        super(name,age,gender,address)
        this.empid = empid
        this.company = company
        this.salary = salary
    }
}

Employee e = new Employee("Rishabh", 22, "Male", "Delhi", 1111, "Rxlogix", 999990)

println "Name: ${e.name}, Company: ${e.company}, EmpID: ${e.empid}, Salary: $
{e.salary}"

println "Name: ${e.getName()}, Company: ${e.getCompany()}, EmpID: ${e.getEmpid()},
Salary: ${e.getSalary()}"

println "Name: ${e.name}, Company: ${e.@company}, EmpID: ${e.@empid}, Salary: $
{e.@salary}"
```



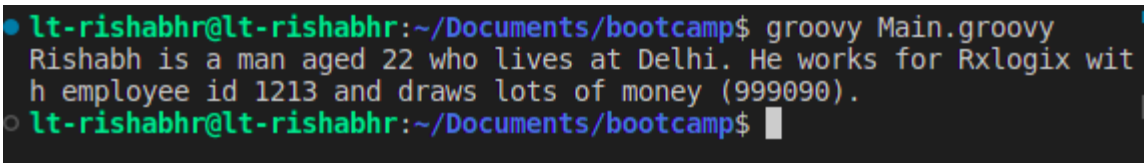
A3.

```
for(i in 0..3){
    (2**i).times{print "*"}
    println "\n"
}
```



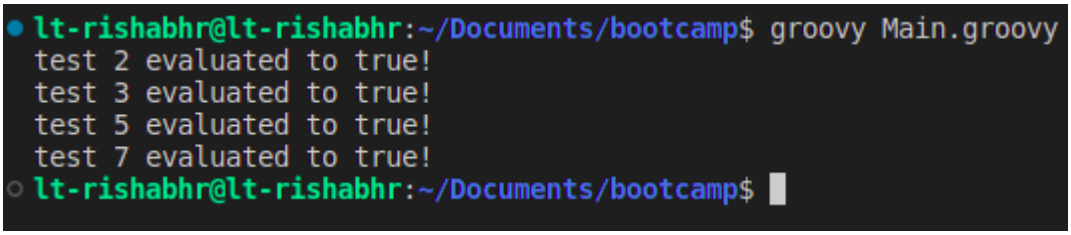
A4.

```
class Employee extends Person{
    Integer empid
    String company
    Integer salary
    Employee(){}
    Employee(String name, Integer age, String gender, String address, Integer
empid, String company, Integer salary){
        super(name,age,gender,address)
        this.empid = empid
        this.company = company
        this.salary = salary
    }

    String toString(){
        return "${this.name} is a man aged ${this.age} who lives at $
{this.address}. He works for ${this.company} with employee id ${this.empid} and
draws lots of money (${this.salary})."
    }
}

Employee e = new Employee("Rishabh", 22, "Male", "Delhi", 1213, "Rxlogix", 999090)
println e.toString()
```



A5.

```
if([]){println "test 1 evaluated to true!"}
if(["true"]){println "test 2 evaluated to true!"}
if(["true","false"]){println "test 3 evaluated to true!"}
if(0){println "test 4 evaluated to true!"}
if(1){println "test 5 evaluated to true!"}
if(false){println "test 6 evaluated to true!"}
if(true){println "test 7 evaluated to true!"}
```



PTO

A6.

```
class HourMinute {
    Integer hour
    Integer minute

    HourMinute minus(HourMinute h){
        Integer total = (this.hour*60+this.minute)-(h.hour*60+h.minute)
        return new HourMinute(hour:total/60, minute:total%60)
    }

    HourMinute plus(HourMinute h){
        Integer total = (this.hour*60+this.minute)+(h.hour*60+h.minute)
        return new HourMinute(hour:total/60, minute:total%60)
    }

    String toString(){
        return "${this.hour}:${this.minute}"
    }
}

HourMinute h1 = new HourMinute(hour:5, minute:15)
HourMinute h2 = new HourMinute(hour:3, minute:55)
HourMinute h3 = h1+h2
HourMinute h4 = h1-h2
println h1.toString()
println h2.toString()
println h3.toString()
println h4.toString()
```

```
lt-rishabhr@lt-rishabhr:~/Documents/bootcamp$ groovy Main.groovy
5:15
3:55
9:10
1:20
lt-rishabhr@lt-rishabhr:~/Documents/bootcamp$
```

A7.

```
for(i in 1..10){
    print "${3*i} "
}

println "\n"
def i=1
10.times{
    print "${3*i++} "
}

println "\n"
[1,2,3,4,5,6,7,8,9,10].each{
    print "${3*it} "
}

println "\n"
```

```
lt-rishabhr@lt-rishabhr:~/Documents/bootcamp$ groovy Main.groovy
3 6 9 12 15 18 21 24 27 30

3 6 9 12 15 18 21 24 27 30

3 6 9 12 15 18 21 24 27 30
```

A8.

```
def containsValue = {value,myList->
    return {
        -> myList.contains(value)
    }
}

def cars = ["bmw", "lexus", "nissan"]
def containsBMW = containsValue('bmw',cars)
def containsMercedeze = containsValue('mercedeze',cars)

if(containsMercedeze()) println "the list contains mercedeze!"
else println "the list does not contain mercedeze!"

if(containsBMW()) println "the list contains bmw!"
else println "the list does not contain bmw!"
```

```
lt-rishabhr@lt-rishabhr:~/Documents/bootcamp$ groovy Main.groovy
the list does not contain mercedeze!
the list contains bmw!
lt-rishabhr@lt-rishabhr:~/Documents/bootcamp$
```

A9.

```
import java.nio.file.Files
import java.nio.file.Paths

def inputDirPath = "/home/lt-rishabhr/Desktop/input/"
def outputFilePath = "/home/lt-rishabhr/Desktop/output.txt"

def inputFiles = new File(inputDirPath).listFiles().findAll { it.isFile()}

def combinedContent = inputFiles.collect {
    Files.readAllLines(Paths.get(it.getPath())).join("\n")
}
.join("\n")

new File(outputFilePath).withWriter {
    writer -> writer.write(combinedContent)
}
```

PTO

```
lt-rishabhr@lt-rishabhr:~/Desktop$ touch output.txt
lt-rishabhr@lt-rishabhr:~/Desktop$ mkdir input
lt-rishabhr@lt-rishabhr:~/Desktop$ cd input
lt-rishabhr@lt-rishabhr:~/Desktop/input$ echo "file1" > file1.txt
lt-rishabhr@lt-rishabhr:~/Desktop/input$ echo "file2" > file2.txt
lt-rishabhr@lt-rishabhr:~/Desktop/input$ echo "file3" > file3.txt
lt-rishabhr@lt-rishabhr:~/Desktop/input$ cat ../output.txt
file1
file3
file2lt-rishabhr@lt-rishabhr:~/Desktop/input$
```

A10.

```
def inputFilePath = "/home/lt-rishabhr/Desktop/input.txt"
def outputFilePath = "/home/lt-rishabhr/Desktop/output.txt"
def oddLines = []
new File(inputFilePath).withReader {reader->
    int lineNum = 1
    reader.eachLine {
        line->
        if(lineNum%2==1) {
            oddLines << "${lineNum}: ${line}"
        }
        lineNum++
    }
}
new File(outputFilePath).withWriter {
    writer ->
    oddLines.each {
        line->
        writer.write(line+"\n")
    }
}
```

```
lt-rishabhr@lt-rishabhr:~/Desktop$ cat input.txt
this is an odd line
this is an even line
this is another odd line
this is another even line
this is third odd line
this is third even line
lt-rishabhr@lt-rishabhr:~/Desktop$ touch output.txt
lt-rishabhr@lt-rishabhr:~/Desktop$ cat output.txt
lt-rishabhr@lt-rishabhr:~/Desktop$ groovy ../Documents/bootcamp/Main.groovy
lt-rishabhr@lt-rishabhr:~/Desktop$ cat output.txt
1: this is an odd line
3: this is another odd line
5: this is third odd line
lt-rishabhr@lt-rishabhr:~/Desktop$
```

A11.

```
def inputFilePath = "/home/lt-rishabhr/Desktop/input.txt"
def outputFilePath = "/home/lt-rishabhr/Desktop/output.txt"
def outputString = ""

new File(inputFilePath).withReader {
    reader->
    reader.eachLine {
        line->
        //remove all whitespaces and append it to the output string
        outputString += line.replaceAll(/\s/,"")
    }
}

new File(outputFilePath).withWriter {
    writer->
    writer.write(outputString)
}
```

```
lt-rishabhr@lt-rishabhr:~/Desktop$ cat input.txt
this is an odd line
this is an even line
this is another odd line
this is another even line
this is third odd line
this is third even line
lt-rishabhr@lt-rishabhr:~/Desktop$ echo "" > output.txt
lt-rishabhr@lt-rishabhr:~/Desktop$ cat output.txt

lt-rishabhr@lt-rishabhr:~/Desktop$ groovy ../Documents/bootcamp/Main.groovy
lt-rishabhr@lt-rishabhr:~/Desktop$ cat output.txt
thisisanoddlinethisisanevenlinethisisanotheroddlinethisisanotherevellltlltll
lt-rishabhr@lt-rishabhr:~/Desktop$
```

A12.

```
def inputFilePath = "/home/lt-rishabhr/Pictures/cars/bmwe46.jpg"
def outputFilePath = "/home/lt-rishabhr/Desktop/img-copy.jpg"

def inputBytes = new File(inputFilePath).getBytes()
new File(outputFilePath).setBytes()
```

```
lt-rishabhr@lt-rishabhr:~$ groovy Documents/bootcamp/Main.groovy
lt-rishabhr@lt-rishabhr:~$ cd Desktop/
lt-rishabhr@lt-rishabhr:~/Desktop$ ls
'images groovy'   link-sharing
 img-copy.jpg     output.txt
 input.txt        rishabh-rajpurohit-groovy-assignment1-solutions.docx
 java-practice
lt-rishabhr@lt-rishabhr:~/Desktop$
```