Introduction to Groovy – Exercise

1. Create a class in Java along with following fields:
   class name: Person
   fields: name, age, gender, address

Access the fields in all known ways (like through getter, by dot operator)

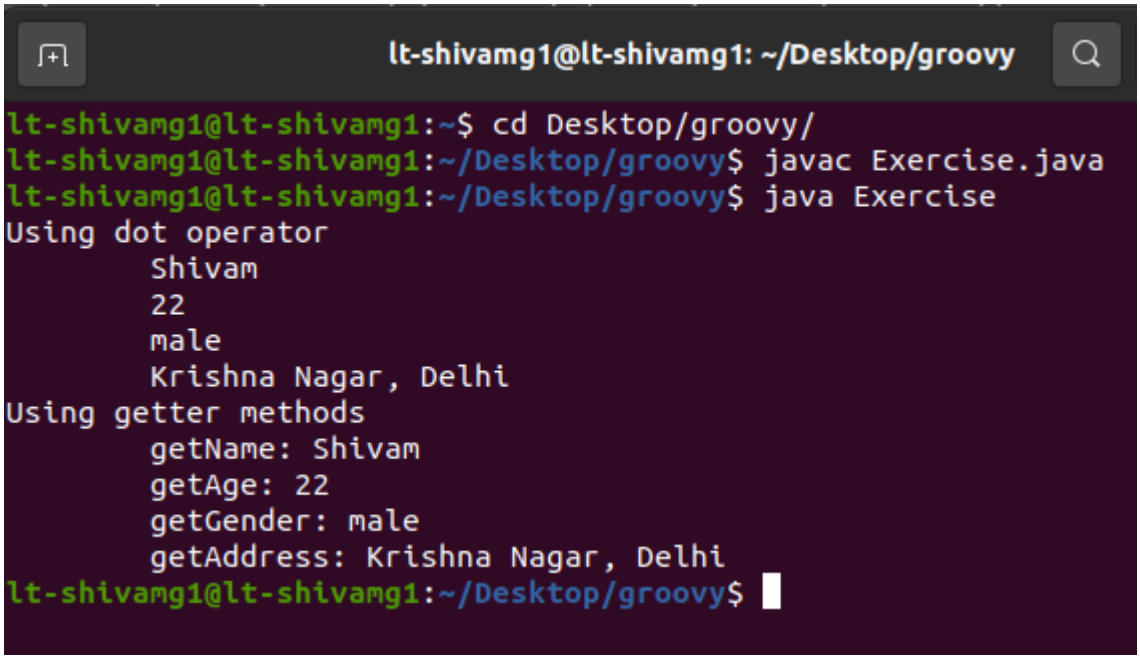
Ans.

Open

Exercise.java
~/Desktop/groovy

```
class Person{
    String name; int age; String gender; String address;
    Person(String name, int age, String gender, String address){
        this.name=name;
        this.age=age;
        this.gender=gender;
        this.address=address;
    }

    String getName(){
        return "getName: " +name;
    }
    String getAge(){
        return "getAge: "+age;
    }
    String getGender(){
        return "getGender: "+gender;
    }
    String getAddress(){
        return "getAddress: "+address;
    }
}
class Exercise{
        public static void main(String args[]){
                Person p1=new Person("Shivam",22,"male","Krishna Nagar, Delhi");
                System.out.println("Using dot operator");
                System.out.println("\t"+p1.name);
                System.out.println("\t"+p1.age);
                System.out.println("\t"+p1.gender);
                System.out.println("\t"+p1.address);
                System.out.println("Using getter methods");
                System.out.println("\t"+p1.getName());
                System.out.println("\t"+p1.getAge());
                System.out.println("\t"+p1.getGender());
                System.out.println("\t"+p1.getAddress());
        }
}
```



2. Extend the Person class in Groovy. Add following fields to it: empId, company, salary. Access the fields in all known ways (like through getter, by dot operator, by @ operator)

Ans.

Test.groovy

```groovy
class Employee extends Person{
        int empId; String company; int salary
        Employee(String name, int age, String gender, String address, int empId, String company, int salary){
                super(name,age,gender,address)
                this.empId=empId
                this.company=company
                this.salary=salary
        }
        String getEmpId(){
                return "getEmpId: ${empId}"
        }
        String getCompany(){
                return "getCompany: ${company}"
        }
        String getSalary(){
                return "getSalary: ${salary}"
        }
}
Employee el=new Employee("Shivam",22,"male","Krishan Nagar, Delhi",42,"RxL",500000)
println """Using dot operator

            Name: ${el.name}
            Age: ${el.age}
            Gender: ${el.gender}
            Address: ${el.address}
            Employee ID: ${el.empId}
            Company: ${el.company}
            Salary: ${el.salary}
"""
println """Using custom getters

            ${el.getName()}
            ${el.getAge()}
            ${el.getGender()}
            ${el.getAddress()}
            ${el.getEmpId()}
            ${el.getCompany()}
            ${el.getSalary()}
"""
println """Using @ operator

            Name: ${el.@name}
            Age: ${el.@age}
            Gender: ${el.@gender}
            Address: ${el.@address}
            Employee ID: ${el.@empId}
            Company: ${el.@company}
            Salary: ${el.@salary}
"""
```

```
lt-shivamg1@lt-shivamg1: ~/Desktop/groovy
lt-shivamg1@lt-shivamg1:~/Desktop/groovy$ groovy Test.groovy
Using dot operator

            Name: getName: Shivam
            Age: getAge: 22
            Gender: getGender: male
            Address: getAddress: Krishan Nagar, Delhi
            Employee ID: getEmpId: 42
            Company: getCompany: RxL
            Salary: getSalary: 500000

Using custom getters

            getName: Shivam
            getAge: 22
            getGender: male
            getAddress: Krishan Nagar, Delhi
            getEmpId: 42
            getCompany: RxL
            getSalary: 500000

Using @ operator

            Name: Shivam
            Age: 22
            Gender: male
            Address: Krishan Nagar, Delhi
            Employee ID: 42
            Company: RxL
            Salary: 500000

Caught: java.lang.NullPointerException: Cannot get property '' on null object
java.lang.NullPointerException: Cannot get property '' on null object
        at Test.run(Test.groovy:46)
lt-shivamg1@lt-shivamg1:~/Desktop/groovy$
```

3. Print this pattern:

```
*
**
****
********
```

Ans.

```
Open
1 for(i in 0..3){
2         a=2**i
3         a.times{print "*"}
4         print "\n"
5 }
```

```
lt-shivamg1@lt-shivamg1: ~/Desktop/groovy
lt-shivamg1@lt-shivamg1:~/Desktop/groovy$ groovy Test.groovy
*
**
****
********
lt-shivamg1@lt-shivamg1:~/Desktop/groovy$
```

4. 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!!!!.”

Ans.

Test.groovy

```
class Employee extends Person{
        int empId
        String company
        int salary
        Employee(String name, int age, String gender, String address, int empId, String company, int salary){
                super(name,age,gender,address)
                this.empId=empId
                this.company=company
                this.salary=salary
        }
        String pronoun
        String str="male"
        void setPronoun(){
                if(gender==str){
                        pronoun="He"
                } else {
                        pronoun="She"
                }
        }
        @Override
        String toString(){
                return """
                ${name} is a ${gender} aged ${age} who lives at ${address}.
                ${pronoun} works for ${company} with employee ID ${empId} and draws ${salary} lots of money!!!!.
                """
        }
}

Employee e1=new Employee("Shivam",22,"male","Krishna Nagar, Delhi",42,"RxL",500000)
Employee e2=new Employee("Shreya",25,"female","Ghaziabad",76,"Acc",750000)
e1.setPronoun()
e2.setPronoun()
println e1.toString()
println e2.toString()
```

```
lt-shivamg1@lt-shivamg1: ~/Desktop/groovy
lt-shivamg1@lt-shivamg1:~/Desktop/groovy$ groovy Test.groovy

                Shivam is a male aged 22 who lives at Krishna Nagar, Delhi.
                He works for RxL with employee ID 42 and draws 500000 lots of money!!!!.


                Shreya is a female aged 25 who lives at Ghaziabad.
                She works for Acc with employee ID 76 and draws 750000 lots of money!!!!.

lt-shivamg1@lt-shivamg1:~/Desktop/groovy$
```

5. Groovy Truth: if(‘test’) {println “test evaluated to true inside if” }

Try replacing test with various objects and observe its behaviour.

a) “Test”
b) empty variable
c) positive number
d) negative number
e) zero
f) filled list
g) empty list
h) list with all values as null List list=new ArrayList()

Ans.

Open

Test.groovy
~/Desktop/groovy

```
String var1="test"
String var2=""
List list1=new ArrayList()
list1=["Shivam",10]
List list2=new ArrayList()
list2=[]
List list3=new ArrayList()
list3=[null,null]

if(var1){println "True for filled variable"}
else {println "False for filled variable"}

if(var2){println "True for empty variable"}
else {println "False for empty variable"}

if(1){println "True for positive number"}
else {println "False for positive number"}

if(-1){println "True for negative number"}
else {println "False for negative number"}

if(0){println "True for zero"}
else {println "False for zero"}

if(list1){println "True for filled list"}
else {println "False for filled list"}

if(list2){println "True for empty list"}
else {println "False for empty list"}

if(list3){println "True for list with only null values"}
else {println "False for list with only null values"}
```

lt-shivamg1@lt-shivamg1: ~/Desktop/g

```
lt-shivamg1@lt-shivamg1:~/Desktop/groovy$ groovy Test.groovy
True for filled variable
False for empty variable
True for positive number
True for negative number
False for zero
True for filled list
False for empty list
True for list with only null values
lt-shivamg1@lt-shivamg1:~/Desktop/groovy$
```

6. Write a class HourMinute where the class stores hours and minutes as separate fields. Overload + and – operator for this class.

Ans.

```
class HourMinute{
        int hour
        int minute
                HourMinute(int hour, int minute){
                this.hour=hour
                this.minute=minute
        }
        def plus(HourMinute obj){
                int var1=this.hour+obj.hour
                int var2=this.minute+obj.minute
                var1+=var2/60
                var2%=60
                return new HourMinute(var1,var2)
        }
        def minus(HourMinute obj){
                int var3,var4
                if(this.minute<obj.minute){
                        var3=this.hour-1
                        var4=this.minute+60
                }
                int var1=var3-obj.hour
                int var2=var4-obj.minute
                var1+=var2/60
                var2%=60
                return new HourMinute(var1,var2)
        }
}

HourMinute hm1=new HourMinute(100,55)
HourMinute hm2=new HourMinute(10,62)
HourMinute hm3=hm1+hm2
HourMinute hm4=hm1-hm2
println "Time 1 -> ${hm1.hour} hr : ${hm1.minute} min"
println "Time 2 -> ${hm2.hour} hr : ${hm2.minute} min"
println "Adding times -> ${hm3.hour} hr : ${hm3.minute} min"
println "Subtracting times -> ${hm4.hour} hr : ${hm4.minute} min"
```

```
lt-shivamg1@lt-shivamg1:~/Desktop/groovy$ groovy Test.groovy
Time 1 -> 100 hr : 55 min
Time 2 -> 10 hr : 62 min
Adding times -> 111 hr : 57 min
Subtracting times -> 89 hr : 53 min
lt-shivamg1@lt-shivamg1:~/Desktop/groovy$
```

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

Ans.

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

println "\nUsing .times method"
int var=3
int var2=2
10.times{print "${var}  "; var=3*var2; var2++}

println "\nUsing .each method"
list=[3,6,9,12,15,18,21,24,27,30]
list.each{print "${it}  "}
print "\n"
```

```
lt-shivamg1@lt-shivamg1:~/Desktop/groovy$ groovy Test.groovy
Using range
3  6  9  12  15  18  21  24  27  30
Using .times method
3  6  9  12  15  18  21  24  27  30
Using .each method
3  6  9  12  15  18  21  24  27  30
lt-shivamg1@lt-shivamg1:~/Desktop/groovy$
```

8. Write a closure which checks if a value is contained within a list where the closure accepts two parameters.

Ans.

```
Closure c={def str, List list->
        if(list.contains(str)){
                println "list contains ${str}"
        } else {
                println "list does not contain ${str}"
        }
}
c.call(10,[1,"gurg",10,"rue"])
c.call(11,[1,"gurg",10,"rue"])
```

```
lt-shivamg1@lt-shivamg1: ~/Desktop
lt-shivamg1@lt-shivamg1:~/Desktop/groovy$ groovy Test.groovy
list contains 10
list does not contain 11
lt-shivamg1@lt-shivamg1:~/Desktop/groovy$
```

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

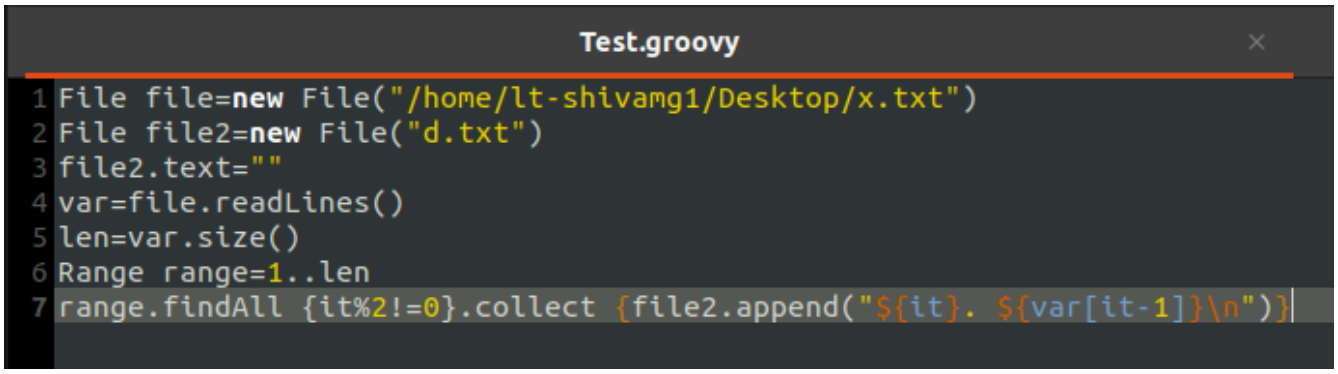
Ans.

```
Open
1 list=["a.txt","b.txt","c.txt"]
2 File file2=new File("d.txt")
3 file2.text=""
4 list.each{
5         File file=new File("/home/lt-shivamg1/Desktop/${it}")
6         var=file.readLines()
7         var.each{
8                 file2.append("${it}\n")
9         }
10 }
```

```
lt-shivamg1@lt-shivamg1:
lt-shivamg1@lt-shivamg1:~/Desktop/groovy$ cat ~/Desktop/a.txt
Hello
This is file
              /\
             /  \
            /____\
           /      \
          /        \
lt-shivamg1@lt-shivamg1:~/Desktop/groovy$ cat ~/Desktop/b.txt
Hello
This is file
       ___
      |   \
      |    |
      |___/
      |   \
      |    |
      |___/
lt-shivamg1@lt-shivamg1:~/Desktop/groovy$ cat ~/Desktop/c.txt
Hello
This is file
       ______
      /      \
     |
     |
     |
     |
      \______/
lt-shivamg1@lt-shivamg1:~/Desktop/groovy$
```

10. 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,.....

Ans.



```
lt-shivamg1@lt-shivamg1: ~/Desktop/groovy
lt-shivamg1@lt-shivamg1:~/Desktop/groovy$ groovy Test.groovy
lt-shivamg1@lt-shivamg1:~/Desktop/groovy$ cat ~/Desktop/x.txt
Hello
Welcome
To
Groovy
!!!!!!


Goodbye
lt-shivamg1@lt-shivamg1:~/Desktop/groovy$ cat d.txt
1. Hello
3. To
5. !!!!!!
7.
lt-shivamg1@lt-shivamg1:~/Desktop/groovy$
```

11. 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.

Ans.

Test.groovy

```
1 File file=new File("/home/lt-shivamg1/Desktop/x.txt")
2 File file2=new File("d.txt")
3 file2.text=""
4 String var=file.readLines().join()
5 file2.append(var.tokenize().join())
6 file2.append("\n")
```

```
lt-shivamg1@lt-shivamg1: ~/Desktop/groovy
lt-shivamg1@lt-shivamg1:~/Desktop/groovy$ groovy Test.groovy
lt-shivamg1@lt-shivamg1:~/Desktop/groovy$ cat ~/Desktop/x.txt
Hello World
        Welcome
To
Groovy
!!!!!!


Goodbye
lt-shivamg1@lt-shivamg1:~/Desktop/groovy$ cat d.txt
HelloWorldWelcomeToGroovy!!!!!!Goodbye
lt-shivamg1@lt-shivamg1:~/Desktop/groovy$
```

12. Make a copy of an image type file byte by byte.

Ans.

Open

Test.groovy
~/Desktop/groovy

```
1 File file=new File("/home/lt-shivamg1/Desktop/mascot-logo-design_fb-img_1200x800.jpg")
2 File file2=new File("/home/lt-shivamg1/Desktop/groovy/copied_image.jpg")
3 var=file.readBytes()
4 file2.setBytes(var)
```

lt-shivamg1@lt-shivamg1: ~/Desktop/groovy

```
lt-shivamg1@lt-shivamg1:~/Desktop/groovy$ groovy Test.groovy
lt-shivamg1@lt-shivamg1:~/Desktop/groovy$
```