TUTORIAL 7

Question 1

1. Describe how the + operator is implemented in the Groovy language for the type String, and the effect of applying + to two String variables.

2. The following method applies the + operator to two parameters

```
def addThem(str1, str2) {
    return str1 + str2
}
```

   What is the value of *added* after the following method call?

```
def added = addThem("Groovy", "Goodness")
```

3. What is the value of *added2* after the following method call?

```
def added2 = addThem(4, 2)
```

4. What characteristics of Groovy's type system allow the method *addThem* to work in both cases?

5. A class WrongNumber is defined as follows

```
class WrongNumber {
    int value

    WrongNumber(int value) {this.value = value}
}
```

   What would need to be added to the WrongNumber class to allow the following assertion to be true (in other words, for the value of *added3* to be equal to 2)?

```
def w1 = new WrongNumber(4)
def w2 = new WrongNumber(2)
def added3 = addThem(w1,w2)
assert added3 == 2
```

Question 2

A class Person is defined in Groovy as follows:

```
class Person {
    String name

    Person(String name) {
        this.name = name
    }

    boolean equals(Object o) {
        System.out.println("Huh! not comparable!")
        return false
    }

    boolean equals(Person emp) {
        return name.equals(emp.name)
    }
}
```

1. What will the output be from the following code?

```
Object e1 = new Person("Alice");
Object e2 = new Person("Bob");
println(e1.equals(e2));
println(e1.equals(e1));
println(e1.equals("Alice"));
```

2. What feature of Groovy does this behaviour illustrate?

3. How does this behaviour differ from Java/C#?

Question 3

A class LanguageList is defined in Groovy as follows:

Note – *name[4..-1]* gives the remainder of the string name after the first 4 characters, e.g. *findGroovy -> Groovy*

```
class LanguageList {
    def list = ['Java', 'Groovy', 'Scala']

    def methodMissing(String name, args) {
        if (name.startsWith("find")) {
            def result = list.find { it == name[4..-1] }
            this.metaClass."$name" = {-> result + "[cache]" }
            result
        } else {
            throw new MissingMethodException(name,
                this.class, args)
        }
    }
}
```

1. Under what circumstances will the *methodMissing* method of LanguageList be invoked?

2. What does the following line of code in *methodMissing* do?

```
this.metaClass."$name" = {-> result + "[cache]" }
```

3. What will be the output from the following code?

```
def languages = new LanguageList()

println(languages.findGroovy())
println(languages.findScala())
println(languages.findJava())
println(languages.findRuby())

println(languages.findGroovy())
println(languages.findScala())
println(languages.findJava())

println(languages.getGroovy())
```

Question 4

Note – *name[4..-1]* gives the remainder of the string name after the first 4 characters, e.g. *findGroovy -> Groovy*

1. The following traits and classes are defined in Groovy:

```
trait Person {
}

trait Teacher extends Person {
    def permissions(){ println("can view and edit module info") }
}

trait Student extends Person {
    def permissions(){ println("can view module info") }
    def results(){ println("exam results...") }
}

class TeachingAssistant implements Teacher, Student {
    String name
}
```

(i) What is the output from the following code?

```
def t = new TeachingAssistant()
t.name = "Alice"
t.results()
t.permissions()
```

(ii) A teaching assistant is a student who does some teaching, so needs to be able to edit module information. Is this the case? If not, identify two different ways in which you could modify the TeachingAssistant class to make sure that permissions are inherited from the Teacher trait.