**Java Quiz: Fundamental Semester**

This Quiz is composed of two parts:

* First part : a list of MCQs, 0 to 2 answers can be given
* Second part : open questions

There is no negative point if you give a wrong answer. Good luck !

# Part 1 : MCQs

1. **How does a Java program run onto the target operating system**

* 1. It uses direct accesses to native target os functions
  2. It uses an abstract virtual machine that runs onto the target os
  3. It has to be pre-compiled then the result is interpreted by the JVM
  4. It doesn't need to be compiled, it runs directly by JVM interpretation

1. **How to define a class in the simplest way?**

* 1. public java class Identity{}
  2. java class Identity {}
  3. java Identity {}
  4. class Identity {}

1. **What is JDBC**

* 1. A protocol to deal with databases communication with the Java language
  2. Java Dust Bin Collector
  3. A protocol that helps to connect to remote file systems
  4. Java Data Base Connectivity

1. **How to define a data base connection through Java ?**

* 1. You need the database connection String
  2. You need the connection string and a Driver
  3. you only need the driver
  4. You need nothing, Java already includes that feature

1. **What is Apache Derby ?** 
   1. a database
   2. an ORM
   3. a Global Object Context
   4. a famous development contest that takes place in the USA

1. **What is/are the concept(s) attached to the java interfaces ?** 
   1. Composition
   2. Inheritance
   3. Realization
   4. Polymorphism

1. **how to test that two Strings, s1 and s2, are equal in Java?** 
   1. s1.equals(s2)
   2. s1 == s2
   3. s1 = s2
   4. s1 :: s2

1. **How many instances does the code below create?**

Thing theThing = new Thing();

Thing anotherThing = theThing;

Thing theThird;

* 1. 0
  2. 1
  3. 2
  4. 3

1. **How do you define an inheritance between two classes?** 
   1. public class A implements B
   2. public class A extends B

1. **Associate keywords with their definitions (note down the number in the parenthesis below)**

* 1. abstract
  2. interface
  3. static
  4. void

( ) defines a behavior or a meta-class, allowing to generify the code and to change the implementation if needed

( ) can be used on class or method definitions, means that a part of the implementation will be assumed by descending classes

( ) can be used on fields or methods, indicating that this field/method is held by the class definition. When placed on a field, the value is shared by all instances of that class

( ) used as a method return type, indicates that the method returns no value.

( ) used on fields or methods, for a field, indicates that the value of that field cannot be changed at runtime. For a method, indicates that the method cannot be overridden by a descending class.

# Part 2

1. Cite two ways of iterating through a List content in java (Java 5 or above)

for (Element el : elements){

el.getTextContent();

}

int size = elements.size();

for (int i = 0 ; i < size ; i++){

Element el = elements.get(i);

el.getTextContent();

}

for (int i = elements.size() - 1; i >= 0; i--){

Element el = elements.get(i);

el.getTextContent();

}

1. Define inheritance, composition, when to favor one or the other?

1. Give a quick Class Diagram that involves all you know about UML Class Diagram notation

1. What is the way to get a JDBC connection using the “org.derby.jdbc.ClientDriver” Driver?

Class.forName(“org.derby.jdbc.ClientDriver”);

String connectionString = “jdbc:derby://localhost:1527/Instance”;

Connection connection = DriverManager.getConnection(connectionString, “user”,”password”);

1. In Eclipse, how to create a new Class?

1. Cite two common Document (class) methods to get XML child nodes using the DOM Apis ?

1. What is the difference between java.lang.Boolean and boolean ?

1. Draw a schema that represents a software design by defining its layers

Datamodel

Java application

User interface

Business Logic

Main

Services

Data access

1. Define the relationship between JRE, JVM, JDK

1. Write a class Identity with a String property (“name”) and an int property (“age”), then initialize it and override the toString() method to display the fields of that class

package fr.tbr.iam.datamodel;

public class Identity{

private String name;

private int age;

public Identity(String name, int age){

this.name = name;

this.age = age;

}

public String toString(){

return this.name + “: “ + this.age;

}

}

package fr.tbr.iam.launcher;

import fr.tbr.iam.datamodel.Identity;

public class Main{

public static void main(String[] args){

Identity identity = new Identity(“Thomas”, 30);

}

}