CentraleSupélec 2017-2018 **Big Data Platforms : Homework 1**

Programming in JAVA : Basics of Object Oriented Programming

The purpose of this homwework is to get you started with JAVA programming and object oriented programming. You will work with the ECLIPSE Integrated Development Environment

Learning outcomes :

— Getting familiar with the ECLIPSE IDE
— Writing your first simple JAVA programs
— Getting familiar with the notion of Java classes and inheritance.

General instructions

— Create a meaningful workspace for the homeworks of the course.
— Within the workspace, create a new package for this homework.
— Carefully comment your code.

Exercice 1 Your first program in JAVA : Hello World

Create a new JAVA project named Howework1 using your IDE.

— In this project, add a new package named `helloworld`.
— Add a new class named `HelloWorld` with a `main` method that displays on the screen the message 'Hello World'.
— Extend the program so that it asks the user to input a (string) message and to display it on the screen.
— Extend the program so that is also displays on screen the value of Π.
— Extend the program so that it asks the user to input an integer value n and then displays on the screen a random number between 0 and n.

Exercice 2 The BIG DATA course

Create a new JAVA project named BIGDATA using your IDE

Question 2.1 Build your first class with constructors and some methods

— Build a `Student` class with the following variables : `name, firstName, average, nbMarks, cursus`
— Add a method `public void addMark(float newMark)` which re-compute the average while adding a mark.
— Add two constructors : without parameters, with two parameters (`name, firstname`).
— Add a method `isPassed` that returns a boolean `True` or `False` is the average of the student is above or below a certain variable `limit`.

Question 2.2 Using Collections

— Build a `Teacher` class with the following variables : `name, firstName, domain, students` with `students` an array of Students.
— Add a public method `boolean acceptStudent (Student aStudent)` which add a Student in the course of the Teacher is not full.

Question 2.3 Inheritance

Add the notion of inheritance in the project by adding a class `Person` that will be the subclass of `Teacher` and `Student`.

Question 2.4 Polymorphism

— Add a method `public String getMail()` that returns the email adress of the person according she is a student (student.centralesupelec.fr) or a teacher (centralesupelec.fr).
— in the class `Person` add a `static` variable (using JAVA Collections) in which all the instances of Person are recorded.
— Add a method `static public String getMailingList()` that returns all the emails of the students and teachers.