Instructions for the programming project

This document is here to explain to you what you will have to do (and how you will do it) for the CS101 programming project in 2014/15 at Mahindra Ecole Centrale.

What to do for this project?

In one sentence : **you (and your team-mates) now have about 2 weeks to write one or more program to solve a scientific problem, as detailed in the other document.**

Be extremely cautious that...

- Cheating will be severely punished. Do not even try to cheat, neither between 2 teams with the same project nor from Internet.
- The deadlines have to be respected! Each day of delay after the 24th will reduce your grade by 4/20.
- **Each member of the team have to participate to the project**.
- Your Python program(s) must work well, and produce correct results, as asked on the project assignment paper you received. The marks for the project team depend significantly on the output and behaviour of the program(s). During the viva part of the project, the team members should demonstrate that the program(s) work as expected.

Important dates to follow

1. Monday 6th of April, you will receive the project paper[1], explaining what you have to work on,
2. During these 3 weeks, you will work regularly with your team-mates and work on your various task, as quickly and efficiently as you can,
3. **Deadline is** Friday 24th of April, you send by email the zip archive (as described below) to the professor in charge of your project (before 21 : 59 pm),
4. During the last week of April (Monday 27th to Friday 1st of May), you will have a short meeting with your professor, to ask *subtle questions* about concepts and *tricky parts* of your code. Each member of the team have to know "everything" about each programs your team have submitted.
5. Grades will be displayed during the first week of May. The best team in each of the 6 projects will get extra marks. *You are competing with the 7 other teams working in the same project*, keep this in mind if you thought of cheating or helping them!

What does your team have to produce and give back?

1. You will have to submit one `zip` archive[2] named `SectionSS__TeamYY__Project_CS101_2015.zip`, with `SS` being the section ($A1$ to $B4$) and `YY` being your project number $(1 \dots 6)$.
2. That `.zip` archive has to be sent by email to the professor in charge of your project (email ID is given on the team list). **Only one submission by team is possible**: you will have to agree on which version you decide to send, and who takes care of it.

[1] There is 6 different projects: 1) FFT, 2) Time Table, 3) Tic-Tac-Toe, 4) Edge Detection, 5) Matrices, or 6) Numerical Integration.

[2] If you do not know how to do that, please read www.wikihow.com/Make-a-Zip-File.

Content of the zip file:

Hint: a complete example of what you have to produce will soon be available on Moodle, based on a "dummy" project (solution of Mid-Term Exam Problem III).

1. One small text file named `AUTHORS.txt` with a list of students in your team, with one name, roll number and email by line, **like this**:

```
Wonder Woman, 14XJ00327, wonder.w@gmail.com
... (2 or 3 more students here) ...
Bat Man, 14XJ00365, bruce@batman.com
```

2. one text file named `LICENSE.txt`, containing the entire copy of the Open-Source license under which your project could be released[3]. This license will have to be chosen in one of these lists: `OpenSource.org/licenses`, or `GNU.org/licenses`. I recommend one of these: the MIT license, the GNU GPLv3 license, the BSD license, or the WFTPL license.
3. one small text file named `README.txt` explaining **in a few lines** what is the project about (what you had to do, *summed up with your own words*),
4. one file named `INSTALL.txt` explaining *briefly* how to use the project. You have to list the modules/packages required by your program (if you used any), and any special manipulations required to use and test your program(s),
5. an *optional* file name `TODO.txt`, that can contain a list of point that still need to be worked on. Can be used to assess how advanced the team was on their subject. And if you tried some bonus questions or extra methods, but have not been able to complete the job, this document will state what would be required to conclude the extra ideas.
6. one file named `SectionSS__TeamYY__Project_CS101_2015.pdf` (**in the PDF format**, not Word `.doc` or `.docx`) containing a **report of your entire work process** (more details are given below),
7. and finally, a list of Python programs (at least one), that solve the problem(s) given to you.

About the report

The goal of such document is for you to be *trained on technical communication*, in order for you to explain what was your task(s); specify what approaches you considered and tried to implement.
Optionally, the report can also contain remarks of any kind about the project (was it interesting? can we improve it for next year? have you tried any extra example?).

Outline ? The suggested outline is the following:

1. Introduction (one paragraph), about the general domain and subject, the task, and the solution that you implemented.
2. The first part can give details about the task(s), what model did you chose, what convention did you follow. The mathematical and scientific notations has to be explained here. Reference books and website can be given here (or at the end of the report),
3. The second part will give details about your Python programs: which function/class does what (in each `.py` file), and how (quickly, no need to write too much). You also have to specify how to use each program (if it is a command line program, is there arguments to give to the script? if not, is there something to change inside the program? If the program is asking any `input` or `raw_input`, are they explicit enough to be understood easily?).

[3] Do not worry, nothing will be publicly released. We just pretend it could be.

4. You also have to clearly say what *non built-in modules*[4] you used. If your program requires any external module that is not included in Anaconda, you need to include a link to the official web-page from where that module(s) can be downloaded (freely).
5. An optional third part If you answered any bonus question or achieved any task additionally to what we ask, please explain what you did, how (and why).
6. Finally, a global conclusion[5], summing up quickly what you learned and what have been achieved.

Tool to write your report ? The suggested way to write this report is either:

- The more common solution: to use Microsoft Word (or any word processing software, like the free clone LibreOffice), and to convert the document to PDF before sending it.
- You can also try to start using LaTeX, the standard for technical and scientific documents. An excellent tool your team could use is the free website `www.ShareLaTeX.com`, to collaboratively write your report,
- You could be interested about the Sphinx documentation generator for Python, if you have time.

Style to follow for your programs:

Conventions

You have to follow the usual Python typing conventions, as we do in lectures, labs and exams, from January. Your programs will be automatically read by the pylint tool (`http://www.pylint.org/`), which assess the quality of you code[6]. `pylint` can be downloaded freely[7] and use easily within Spyder.

`@author` in the doctstring of each program As Spyder is doing it automatically, we ask you to include a line like this one in each of the *docstring* (the big string between triple quotes `"""`) in the beginning of each Python program you submit[8]: `@author: Team 1 Section A3`

Last advices

- Stay in touch with your professor, and contact him if needed.
- Do not mess up this project, its is the only serious programming task of your first year!
- Try to enjoy the project, and work efficiently with your friends.
- DO NOT CHEAT ! You are competing with the other teams (there will be prizes for the best projects!).
- Respect the deadlines, any delay will be severely punished.
- Good luck :)

[4] Remark: each of the 6 project can (and should) be done without using any extra package that not included in Anaconda (for some project, you can use `numpy`, `scipy`, `matplotlib`).

[5] You can send a personal conclusion directly to your professor if you want.

[6] Warning: a too important number of syntax or style warnings or errors in one of your Python program might reduce your final mark for the project.

[7] On Windows, type `conda install pylint` or `pip install pylint` on a DOS console, *when being connected to Internet.*

[8] In case one file is lost, we need to know to whom it belongs.