1IN3000 – Coding Weeks

Instructors: Paolo Ballarini, Marc-Antoine Weisser
Department: DÉPARTEMENT INFORMATIQUE
Language of instruction: ANGLAIS, FRANCAIS
Campus: CAMPUS DE METZ, CAMPUS DE PARIS - SACLAY, CAMPUS DE RENNES
Workload (HEE): 80
On-site hours (HPE): 54,00

Description

The objective of this module is to enable students to consolidate their knowledge in programming and software engineering through coding projects, on the one hand, and, on the other hand, to get familiar with the practices and methodologies of collaborative computer science projects development within a group of developers. Following a "learning-by-doing" approach, students will be initiated to the fundamentals of software craftsmanship.

Quarter number

Infill week at the end of SG1

Prerequisites (in terms of CS courses)

nformation systems and Programming (ISP)

Syllabus

This module is organized in two consecutive weeks of programming in bootcamp mode.

During a first part (3 days), students will work in pairs and groups of 4 to 5 persons to develop a typical project organized in different progressive iterations (sprints). The pairs could be renewed every day (i.e. "pair programming", one of the computer development methods considered as a good practice of "extreme programming"). Short methodological and technical contributions will also be provided in the form of conferences and mini-tutorials (version managers, APIs, Test Driven Development and code quality, agility) and implemented during the development of this model project. This first part will give rise to an evaluation taking into account a knowledge test of the concepts introduced during this first part, a presence test and a participation mark.

During the second part (5 days), students will work in groups of 4 to 5 on a thematic project with the objective of moving from idea to product. The aim will be to implement the previously acquired methodologies and practices. This project will give rise to a group evaluation by a jury during a defense.

Class components (lecture, labs, etc.)

This module is organised over two consecutive weeks
=========
Week 1 - Day 1: 1h30 amphi (presentation and skills input) + EL/TP work in groups of 25-35
Week 1 - Day 2: EL/TP work in groups of 25-35
Week 1 - Day 3: EL/TP work in groups of 25-35
Week 1 - Day 4: work on the project
============
Week 2 - Day 1: work on the project
Week 2 - Day 2: work on the project
Week 2 - Day 3: work on the project
Week 2 - Day 4: work on the project - Final defense during the last half-day.

Grading

The first three learning outcomes will be assessed both by the work on the typical project in the first part (evaluation in the form of mini qcm - 40% of the final grade) and by the work on the free project in week 2 (defence - delivery and documentation of the project on Gitlab or Github - 60% of the final grade).The fourth learning outcome will be more specifically assessed in the part 2 project. In case of a justified absence to one of the intermediary examinations, the grade of this latter is replaced by the grade of the final examination.

Course support, bibliography

1. Programming Languages
   a. Python
      i. Think Python 2nd Edition - Allen B. Downey - GreenTeaPress.
      ii. Python Cookbook, 3rd Edition - Recipes for Mastering Python- B. Jones - D.Beazley - O'Reilly
      iii. Test-Driven Development With Python: Obey the Testing Goat- Harry J.W. Percival- O'Reilly
      iv. Autres : RealPython : https://realpython.com/
   b. Java
      - Core Java Volume I—Fundamentals, Eleventh Edition - Cay S. Horstmann. Prentice Hall

- Software craftmanship

- Clean Code - A Handbook of Agile Software Craftmanship- Robert C. Martin
- Apprenticeship Patterns: Guidance for the Aspiring Software Craftsman- Dave Hoover et Adewale Oshineye - O'Reilly
- OCTO - Culture Code - https://www.octo.com/publications/culture-code/

Resources

- Teaching Team : Paolo Ballarini (responsable), Marc-Antoine Weisser (responsable), Lina Yé, Gianluca Quercini, Wassila Ouerdane, Myriam Tami and a team of temporary staff.
- Group size: 25 to 35 students.
- Programming languages: mainly Python and its many modules and according to groups: java, unity...
- Software tools: VisualStudioCode, git and for some groups Android Studio, Unity

Learning outcomes covered on the course

Through this module the student will be able to:

- Know and Apply good programming practices and methods of software development at a simple project level: structuration of a set of programs in a clear and meaningful way, use of existing libraries or modules, build the project in a modular way, test, and allow the readability, understanding and use of the project source code (software quality).
- Know how to build a coding project in an iterative (from idea to product) and collaborative approach.
- Know and know how to use the main tools of software development.
- Work in team/collaboration.

Description of the skills acquired at the end of the course

- This course will provide a first milestone in competency **C6: " Be operational, responsible and innovative in the digital world**"
- Knowing and applying good programming practices and methods of computer development at the scale of a simple project is covered in **C6.2: "Software Design" and C6.3: "Data Processing".**
- Knowing how to build a computer project in an iterative (from idea to product) is part of **C6.1: "Solve a problem numerically"**
- Knowing and knowing how to use the main tools of software development is part of C6.2: "Software Design"
- Teamwork/collaboration is part of **C8: "Leading a project, a team".**
- Skills **C1.1, C3 and C7** will also be mobilized.

LANGUAGE AND CULTURE COURSES