HOGESCHOOL ROTTERDAM / CMI

Analysis 3
Object-oriented Modeling
(formerly know as Algorithmic Thinking)

INFANL03-3

Study points: 4 ECTS
Course Owner: Jos Vrancken, vrajl@hr.nl room: H.4.206

Course Description

Course Name:	Analysis 3, Object-oriented Modeling (formerly know as Algorithmic Thinking).
Course code:	**INFANL03-3**
Number of studypoints and study load in hours:	4 ECTS (=112 hours) • Contact hours: 4 x 50 minutes per week, 8 weeks = 27 hours • Hand-in exercises: 5 hours per week, 8 weeks = 40 hours • Self study: 45 hours.
Required entry level knowledge:	Secondary school or MBO, Analysis 1 and 2.
Work forms:	Lectures. Supervised practicum. Hand-in assignment in small teams of 1 - 3 students.
Assessment:	• Exam: Multiple Choice exam. Determines final grade. • Practicum: The final assignment (which is a build-up from previous assignments) must be Sufficient, in order to pass the course.
Learning aids:	• Students' Reading Guide: main access point for all documents, code, etc., for this course. • Book: Alex Martelli & et al.: Python in a Nutshell, third edition. • Russ Miles, Kim Hamilton: Learning UML 2.0. • Slides of the lectures • List of Subjects (summary of the content material) • Sets of exercises.
Contributes to competencies:	Analysis, Design, Realisation, all at level 1. (Hogeschoolgids CMI, article 10.2.1)
Learning Objectives:	After this course, the student is capable, at an elementary level, of: **LO1.** Describing the key concepts of Object-orientation and how they are used in modeling and programming. **LO2.** Modeling real-world problems in terms of classes, objects, state and methods and translating this model into an executable solution in Python. **LO3.** Applying existing classes in the Python Standard Library. **LO4.** Expressing models in UML Class and State Transition Diagrams **LO5.** Can model real world objects as Finite State Machines (FSM), when appropriate **LO6.** Can apply FSM to the problem of parsing formal texts, such as regular expressions and JSON-files. OO-concepts, OO-modeling and the FSM-model contribute to the competencies of Analysis and Design. Knowledge of Python contributes to Realisation.
Content:	Object-orientation, OO-modeling, modeling in UML, OO-programming in Python, Finite State Machines, parsing, regular expressions and the JSON-format.
Remarks:	There will be weekly meetings of the teachers involved in the course.

Course manager:	Jos Vrancken
Date:	Version of January 30, 2019

1. General Description

This course is the third step in the Analysis line of education, for the Software Engineering program.

1.1 Introduction

Object-orientation is a powerful concept in modeling and programming. For some well-known languages, such as Java and C#, it is even an exclusive concept: they are meant to be used only in the object-oriented way. The great advantage of object-orientation is that the classes and objects recognized in a real-world problem can be translated into an OO-design and implemented in a OO-language, with the real-world objects still recognizable in the code. This makes object-orientation a concept that serves all phases in the development cycle.

1.2 Relationship with other courses

Other courses in the other learning lines, focus on realization of IT-solutions and on the skills needed to properly playing your role, as software engineer, in projects with people with various other backgrounds and fields of expertise. This course focusses on a modeling concept: it extends your modeling toolkit with the most powerful modeling concept to date: object-orientation. It also shows how to implement object-oriented models in Python, whereas in other courses, you will learn other languages.

1.3 Learning aids

Your main learning aid is a laptop. Other learning aids are mentioned in the Students' Reading Guide, the List of Subjects, course slides, sets of exercises, and free downloadable documents and systems from the Internet, such as the Python programming environment.

2. Program

The table belows shows the learning material per week for the 7 weeks of the course.

Week	Lesson content	Products
1	• Introduction to Object-orientation	In-class exercises
2	• Class diagrams in UML	In-class exercises
3	• Finite State Machines (FSM) and the State Transition Diagram in UML	In-class exercises
4	• Formal languages and grammars. Parsing with FSM	In-class exercises
5	• The JSON format, applications of JSON and JSON parsing	In-class exercises
6	• Storing and Retrieving Data in JSON format. Using data sets from the Internet	Assignment
7	• Rehearsal and Exam Training.	In-class exercises

3. Assessment

3.1 Procedure

The assessment will consist of a summative (graded) hand-in assignment and a written Multiple-Choice exam (40 questions with 4 options per question) in week 9 of the third period. The grade will be determined by the written exam, but the hand-in assignment must be graded as Sufficient, in order to make the grade valid.
During the first period, you will receive feedback on the in-class exercises and the assignment.
A training exam will be made available during the period.

For the written exam, there is a retake in week 8 of the fourth period. For the assignment, there are deadlines for the first attempt (end of week 9 of period 3) and for the second attempt (end of week 8 of period 4). Grading takes place within 15 working days after the deadline.

The written exam and the assignment are two independent components of the assessment.

3.2. Assessment Matrix

The following table shows which learning goals are assessed in either the written exam or the practicum assignment.

	Written exam	Practicum Assignment
LO1: Object-oriented concepts	X	
LO2: Object-oriented modeling	X	
LO3: Implementation in Python	X	X
LO4: Modeling in UML	X	
LO5: Finite State Machines	X	X
LO6: Parsing	X	X