

ROTTERDAM UNIVERSITY OF APPLIED SCIENCES / CMI

Course Manual Development 3

INFDEV02-3

ECTS: 4
Course facilitator: Deborah Barnett

Description of the course

Course name:	Development 3																
Course code:	INFDEV02-3																
Study points and workload:	4 ECTS																
	<p>This course provides the student with 4 ECTS, which corresponds to a study load of 112 hours. The distribution of these 112 hours over the college weeks is as follows:</p> <table> <tr> <td>Guided lectures</td><td></td></tr> <tr> <td>during 7 weeks: 7 × 100 minutes:</td><td>~12 hours</td></tr> <tr> <td>Practicums</td><td></td></tr> <tr> <td>during 7 weeks: 7 × 100 minutes:</td><td>~12 hours</td></tr> <tr> <td>Unaccompanied hours per week</td><td></td></tr> <tr> <td>Self-study (~11 hours per week):</td><td>~78 hours</td></tr> <tr> <td>Test preparation and participation</td><td>10 hours</td></tr> <tr> <td>Total</td><td>112 hours</td></tr> </table>	Guided lectures		during 7 weeks: 7 × 100 minutes:	~12 hours	Practicums		during 7 weeks: 7 × 100 minutes:	~12 hours	Unaccompanied hours per week		Self-study (~11 hours per week):	~78 hours	Test preparation and participation	10 hours	Total	112 hours
Guided lectures																	
during 7 weeks: 7 × 100 minutes:	~12 hours																
Practicums																	
during 7 weeks: 7 × 100 minutes:	~12 hours																
Unaccompanied hours per week																	
Self-study (~11 hours per week):	~78 hours																
Test preparation and participation	10 hours																
Total	112 hours																
Learning goals:	<p>The course has the following learning goals:</p> <ul style="list-style-type: none"> • (ICODE) students can <i>recognize and formally interpret</i> components of a programming language in their formal semantics and type-checking rules: from basic constructs to simple classes; • (ECODE) students <i>understand code</i>; <i>understand</i> here means that students can describe the effect of compiling and running the program by predicting the sequence of typings and state transitions produced by its execution; • (WCODE) students can <i>write small snippets of code</i> in order to complete an existing program, from a clearly given specification of the typings and state transitions that the completed program is expected to perform. <p>The course, and therefore also the learning goals, are limited to imperative, functional, and basic object-oriented programs, without generics, inheritance, polymorphism, or exceptions.</p> <p>All concepts and keywords learnt during the previous development courses (<i>Development 1</i> and <i>Development 2</i>) are assumed to be available and built upon.</p> <p>The corresponding competences connected to these learning goals are:</p> <ul style="list-style-type: none"> • Realisation. 																
Course facilitator:	Deborah Barnett																
Date:	11-2-2019																

1 General Information

Development 3 covers the concepts in and around basic object oriented programming within the boundaries of type safety.

The course opens with a description of the mechanisms of type safety and type checking as a simulation of the correctness and structural integrity of a program. It then moves on to an exploration of the language constructs available in C#, and how those differ from those we have seen in Python (imperative, procedural, and functional constructs). Each language construct is now explored in terms of semantics, but also type-checking. Type-checking is formally seen as a form of semantics which produces a sequence of typings instead of memory states. After exploring the already known aspects of the language, we move on to a presentation of the heap, arrays, and basic classes. We conclude the course with a discussion about the design of classes, and the principles of encapsulation and information hiding.

Many concrete examples are shown throughout the whole course: given the abstract nature of the concepts, we will strive towards showing many instances of these topics in action. The examples are small in size, but complex and exemplary in nature.

An important reminder: the course (and the whole Informatica degree of Hogeschool Rotterdam as well, for that matter) is not meant to provide students with a series of tips and tricks to be quickly successful at one's job. Rather, the course aims to build a solid foundation upon which learning of all sorts of programming languages will be based. This will, without a doubt, prepare students not only for their first professional tasks, but for all their evolutions. Moreover, understanding of these concepts will allow students to wield their professional tools (programming languages) with certainty instead of intuition, thereby greatly increasing the quality of what they deliver and their value on the job market. Finally, the focus on C# rather than, for example, Java, is purely cosmetic. C# offers some aspects which are favorable for beginners (lambda's, tuples, reified generics, etc.) but is not really a structurally better or more useful language with respect to very good alternatives such as Java.

2 Program

The course is made up of a series of lectures and practicums, usually planned as one lecture and one practicum per lesson week.

The lectures begin with an introduction of the basic concepts, which are then elaborated during the practicums. The lectures handle the theory, plus applied examples. Some of the applied examples are then further elaborated during the practicums.

The topics covered by the course are the following:

- Introduction to type-checking, primitive types, tuples, and records;
- Functions, anonymous functions, currying, and closures;
- The heap and references;
- Arrays;
- Basic classes: fields, methods, constructors, references (type-checking and semantics);
- Principles of object-oriented design and encapsulation.

Note that each set of topics does not necessarily correspond to one lesson week (for example, one set of topics could span during two lesson weeks or two sets of topics could be handled during one lesson week).

3 Attendance

Attendance is strongly advised for both the theory and practicum lessons, therefore attendance will be monitored throughout the course.

4 Deliverables

None.

5 Evaluation

See Osiris.