**CO7207 Generative Development**

**Credits:** 15  
**Convenor:** Dr. Artur Boronat  
**Semester:** 2nd

---

**Prerequisites:** Desirable: UML, Java, Eclipse

**Assessment:**  
Coursework: 40%  
Two hour exam: 60%

**Lectures:** 24 hours  
**Surgeries:** 8 hours  
**Laboratories:** 8 hours  
**Private Study:** 72.5 hours

---

**Subject Knowledge**

**Aims**  
The functionality and size of software systems grow in time. This requires more adequate methods for system development. This module covers the main principles and techniques of generative development. It focuses on system modelling and code generation. The module also covers the foundations of aspect-oriented programming (AOP).

**Learning Outcomes**  
At the end of this course, successful students will be able to: be aware of the main approaches for automating software development; critically evaluate the role of modelling and code generation in software development; use UML and OCL for designing views of software systems; check the consistency of the UML design of an application; use techniques for model-driven development; explain concepts of aspect-oriented programming and apply them.

**Methods**  
Class sessions together with course notes, textbooks, printed solutions, and some additional hand-outs and web support.

**Assessment**  
Assessed coursework, traditional written examination.

---

**Skills**

**Aims**  
To teach students abstraction and higher-level modelling skills and to provide the basic skills required to use generative development methods. In particular model-driven development techniques and aspect-oriented programming.

**Learning Outcomes**  
At the end of this course, successful students will be able to: solve abstract and concrete problems (both routine seen, and simple unseen); model and specify software systems; use state of the art tools for code generation; use aspect-oriented programming; develop software in a systematic, automated way.

**Methods**  
Class sessions together with worksheets and practical programming experience.

**Assessment**  
Computer-based exercises, traditional written examination.

---

**Explanation of Prerequisites**  
Basic knowledge of UML and Java is desirable.

**Course Description**  
Software engineering is a very dynamically developing discipline. There are new specification, modelling and programming languages, new tools and paradigms for development of software systems. To the most promising new ideas in recent years are:

- **UML** for modelling of software systems
- **Generative methods** for code generation
- **Aspect-oriented programming** for compositional development of complex systems
- **Model-driven development** for software system development, e.g., OMG’s Model-Driven Architecture (MDA) initiative.
Detailed Syllabus

The course will provide a broad picture of new developments in the area of modelling and code generation. It will teach methods of proper system modelling using UML diagrams, methodical system development from UML model to implementation using generative methods, the principles of Aspect Oriented Programming and MDA. In this course we will use state of the art software tools.

Reading List


Resources Study guide, worksheets, lecture rooms with data projector, computer lab access, handouts.

Module Evaluation Course questionnaires, course review.