

Description of Achievements

Kiko Fernandez-Reyes

0.1 Methodology

This document describes the expanded achievements for the course Advanced Software Design. For each achievement, there is a specification of its level and an indication of how it links with course objectives. You will get continuous feedback on the achievements during the course and present a final document that clearly marks and explains where each achievement has been solved. To get grade 3 on the course, all the achievements at level 3 must be explained in the document. For grade 4, you need to explained all the achievements at level 3 and 4. For grade 5, demonstrate that you have learnt advanced design principles and finish the single achievement of level 5. This achievement is done **individually** and group work is not allowed. Note that achieving higher levels does not entail doing more work per se, as many achievements can be proved at the same time. So, be clever about planning your efforts.

The following are the course objectives, coded from IOL-A–IOL-F. These codes appear next to each achievement to indicate how they relate to the course objectives.

ILO-A Summarise and explain relevant design principles.

ILO-B Use an object-oriented methodology to design and implement larger programs.

ILO-C Use a modelling language as a means to communicate realistic problems and their solutions.

ILO-D Demonstrate understanding of the connection between modelling languages and programming languages, for example by implementing design models.

ILO-E Use design patterns and other known solutions to design problems.

ILO-F Evaluate the suitability of different design alternatives based on object oriented design principles, and identify design flaws in programs.

ILO-Extra Conjunction of ILO-A, ILO-B, ILO-C, ILO-D and ILO-F.

Each achievement has the form:

0-1	Goal	Level	ILO
	A description of the achievement.	L	ILO-O

Criteria. The assessment criteria.

Documents. The documentation you need to provide.

Level represents the grade level of the achievement.

IOL represents the corresponding intended learning outcome.

The following table capture how the achievements are distributed, both across the different grade levels and per course objective.

Course Objective	ILO-A	ILO-B	ILO-C	ILO-D	ILO-E	ILO-F	ILO-Extra	ILO-P	Total
Level 3	1	2	3	2	1	1	-	1	11
Level 4	1	2	3	0	1	3	-	-	10
Level 5	-	-	-	-	-	-	1	-	1
Total	2	4	6	2	2	4	1	1	22

P indicates *procedural objectives* which are in place to ensure timely completion of design tasks.

Meta-instructions

The remainder of this document describes all achievements. Please make a clear distinction between the design tasks you need to perform and these achievements: the specification of the project (the other document) describes what needs to be designed, whereas the achievements (this document) are what need to be demonstrated in the final document to pass the course. The project provides the material upon which to base your demonstrations.

An important insight that you must reach in order to cope with this course is that the achievements are heavily related to the design tasks, and that some are easier if they are solved together. Ultimately, it should be clear that a higher grade is not about quantity, but about synthesis and deeper levels of insight (quality).

Here are a number of tips to guide you:

1. Develop a plan for how to address the achievements, globally.
2. TAs will take note of the interaction and this give us insights to which members are doing the work. The TAs will not (indeed cannot) extract the knowledge from you.
3. Try to get as much feedback from the TA as possible, and focus on the questions where you are stuck.

All members must be present during the feedback moments. Failing to attend will impact your individual grade.

The last date to send the report (Goal K) with all achievements is midday (Central European Time, *CET*) January 10, 2020

The Achievements

A Design Principles

A1	Goal	Level	ILO
	Illustrate the GRASP principles using concepts from your design.	3	ILO-A

Criteria. Answer correctly identifies each GRASP principle using concepts from the system under design.

Documents. Some relevant diagrams, such as class diagrams, possibly independent of your ongoing design.

A2	Goal	Level	ILO
	Explain how GRASP design principles have been applied in the system under design, using compelling examples.	4	ILO-A

Criteria. Answer correctly applied GRASP principles to the existing design. Note that your design needs to be sufficiently sophisticated before you can attempt this achievement.

Documents. Class diagrams from the system-under-design.

B Domain Modelling

B1	Goal	Level	ILO
	Construct a valid domain model of the core elements of the software under design.	3	ILO-C

Criteria. Answer includes the most important domain elements and most important relationships between them, with few spurious elements and relationships, for **Level R** of the specification.

Documents. Domain model.

B2	Goal	Level	ILO
	Construct a valid domain model of a substantial subset of the software under design.	4	ILO-C

Criteria. Answer includes all important domain elements, several extensions, and the relationships between them (**Level R and B** of the specification). An extension is a configuration.

Documents. Domain model.

C Requirements and Software Architecture

C1	Goal	Level	ILO
	Compare and contrast functional and non-functional requirements using examples from the system-under-design, and discuss the purpose of software architecture.	3	ILO-B

Criteria. Answer concisely and correctly identifies various functional and non-functional requirements, and formulates a general characterisation of the difference between the two. Answer also concisely accounts for why it is beneficial to consider software design from an architectural perspective.

Documents. A short document listing the relevant non-functional requirements for the system-under-design.

C2	Goal	Level	ILO
	Analyse the non-functional requirements of the software system under design and produce an appropriate software architecture for the most pressing requirements.	4	ILO-B

Criteria. Answer correctly identifies the non-functional requirements for the system under design, without any spurious requirements, and produces a software architecture in terms of class diagrams and component diagrams, which addresses those requirements.

Documents. Class diagrams and Component diagram. Optional: other architectural design notation.

D Design

D1	Goal	Level	ILO
	Construct appropriate class models of the core of the software under design.	3	ILO-C

Criteria. Answer includes the most important ingredients using syntactically valid UML and covers all core elements. (**Level R.**)

Documents. Domain model, Class diagrams.

D2	Goal	Level	ILO
	Construct comprehensive class models of the software under design.	4	ILO-C

Criteria. Answer includes the most important ingredients using syntactically valid UML, and covers all core elements and some extensions in detail. (**Level R and B.**)

Documents. Domain model, Class diagrams.

	Goal	Level	ILO
D3	Provide behavioural models that detail the basic transitions within the system.	3	ILO-C

Criteria. Answer includes behavioural models of the core part of the system. (**Level R.**)

Documents. Class diagrams, Behavioural models such as sequence diagrams.

	Goal	Level	ILO
D4	Construct comprehensive behavioural models for one end-to-end configuration of the system under design.	4	ILO-C

Criteria. Answer include behavioural models of the subsystem in detail. (**Level B.**)

Documents. Class diagrams, Behavioural models such as sequence diagrams.

E Implementation

	Goal	Level	ILO
E1	Demonstrate the connection between class diagrams and code, motivating any deviations between design and code, using examples from the system under design.	3	ILO-D

Criteria. Answer identifies UML elements and corresponding code in the language of your choice, without spurious connections, covering classes, inheritance, encapsulation, associations, etc. Differences are accounted for concisely and correctly. The program does not need to run, but it should compile without any warning or errors (preferably a static language)

Documents. Class diagrams, Behavioural diagrams and Code. Include a Makefile to easily compile your code.

	Goal	Level	ILO
E2	Demonstrate the connection between behavioral diagrams and code, motivating any deviations between design and code, using examples from the system under design.	3	ILO-D

Criteria. Answer identifies UML elements and corresponding code in the language of your choice, without spurious connections, covering classes, inheritance, encapsulation, associations, etc. and behavioural models. Differences are accounted for concisely and correctly. The program does not need to run, but it should compile without any warning or errors (preferably a static language)

Documents. Class diagrams, Behavioural diagrams and Code. Include a Makefile to easily compile your code.

F Design Patterns

	Goal	Level	ILO
F1	Identify 2 or 3 non-trivial design problems within the system under design which could be solved using non-trivial design patterns.	3	ILO-E

Criteria. Answer correctly identifies design problem and which pattern(s) could address the problem, without playing *pattern bingo*, that is, without guessing and hoping for the best. Trivial patterns such as *singleton* and *wrappers* cannot be considered.

Documents. Documents with concise explanation.

	Goal	Level	ILO
F2	Apply the design patterns to the problems identified in F1 .	4	ILO-E

Criteria. Answer correctly applies design patterns without introducing spurious elements, and justifies the choices made (justify the selected pattern, justify how applied). Check with your TA that you've selected a sufficiently complex design problem.

Documents. Class diagrams, Behaviour diagrams (sequence diagrams).

G Design Improvement and Refactoring

	Goal	Level	ILO
G1	Incorporate substantial new use case into a design.	3	ILO-B

Criteria. Answer shows design documentation before and after the change, highlighting where changes have been made. New use case must be non-trivial, and changes correctly capture the new requirements.

Documents. Domain model, Class diagrams, Behavioural diagrams. Description of use case.

	Goal	Level	ILO
G2	Refactor design to improve it to overcome identified weaknesses. Weakness will be determined in conjunction with TA.	4	ILO-F

Criteria. Answer includes both original and refactored design with refactoring steps clearly noted. Each refactoring is justified according to the design principles, and the refactoring must arguably be an improvement in the design.

Documents. All design documents.

	Goal	Level	ILO
G3	Estimate how a request for changes to the software-under-design will affect it and develop a plan for implementing those changes. Requirements will be provided by TA.	4	ILO-F

Criteria. Answer is realistic and identifies most or all of the places in which the software needs to

change, including any refactoring required, and what those changes entail. All proposed changes are justified based on design principles. Plan is well-thought-out. Design needs to be sufficiently advanced before attempting this.

Documents. All design documents.

H Design Review

	Goal	Level	ILO
H1	Objectively review own final design. Identify weaknesses and reason about completeness of various use cases. Use GRASP and walk-throughs.	3	ILO-F

Criteria. Answer systematically and concisely applies GRASP principles and walk-throughs on own design and identifies weaknesses and justifies whether use cases are complete. You will need to clearly give the use cases being considered, using the project description as a basis.

Documents. All design documents and a report that concisely identifies weaknesses and strengths of your design. Link this report with the design documents.

	Goal	Level	ILO
H2	Assess the quality of another group's design in relation to the specification. Identifying both positive and negative attributes based on GRASP principles, walk-throughs, and other methods.	4	ILO-F

Criteria. Answer systematically and concisely applies GRASP principles and walk-throughs on other design and identifies some weaknesses and justifies whether some use cases are complete or not. At least 3 non-trivial use cases are walked-through and 4 GRASP principles are applied. (**This achievement can be done only in week 50 – 51. If your team would like to work on this before the date, please contact the main lecturer and TA.**).

Documents. Design documents from another team **to be provided**.

I Teamwork

	Goal	Level	ILO
I1	Enumerate difficulties your team has encountered throughout the course and propose reasons for those difficulties.	3	ILO-B

Criteria. Answer concisely identifies a number of difficulties and provides a reasonable account of the cause of them (without attributing blame). **This can only be done towards the end of the course.**

Documents. None.

J Individual work

	Goal	Level	ILO
J1	Construct appropriate domain, class and behaviour model for an application of your choosing. Consider the functional and non-functional requirements (agreed with your TA) and produce an appropriate software architecture. This achievement can be delivered until January 10, 2020 upon agreement with TA.	5	ILO-Extra

Criteria. Answer concisely identifies important domain elements, classes, behavioural models, software architecture and applied design patterns. The student can continue working in the current design if so desired, extending the design with **Level X** features.

Documents. Class diagrams, Behavioural models, Architectural design, Functional and Non-functional requirements and explanation of used design patterns.

K Deliverables

	Goal	Level	ILO
K1	Produce comprehensible documentation of design and modelling artefacts, and bundle these together in a final deliverable.	3	ILO-P

Criteria. A significant volume of design documents are submitted to the TAs in time for the review.

Documents. All design documents.

Total number of achievements 22.