Object Oriented Analysis and Design Student Manual

version 0.07

Riemer van Rozen

Hogeschool van Amsterdam

Wibautstraat 2-4, 1091, GM Amsterdam

October 13, 2015

Abstract

This is a manual to the course Object Oriented Analysis and Design. It describes learning goals, a week-by-week course schedule, obligatory assignments and grading. In a nutshell, the manual explains how to pass this course. Updates of this document are provided on the Virtuele Leer-Omgeving (VLO)[1].

1 Course Overview

The course Object Oriented Analysis and Design (OOAD) is a second year bachelor course at the Amsterdam University of Applied Sciences. We provide descriptions of course material in Section 1.1, required course activities in Section 1.2 and a detailed schedule in Section 1.3. Section 2 describes the practical assignments. Please read this manual carefully!

---

[1] https://vlo.informatica.hva.nl/courses/OOAD/

1.1 Study Material

Book. The textbook for OOAD is: Craig Larman. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition)*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2004. ISBN 0-13-148906-2
Slides. The slides are provided on the VLO[2].

1.2 Required Course Activities

- **Homework: Read homework.** Students study a chapter each week, as well as the slides that accompany the chapter. Please check the course schedule for detailed information in Section 1.3.

- **Lecture Part 1 - Seminar: Discuss homework.** During the first part of the lecture the lecturer discusses homework with students and answers questions using the slides.

- **Lecture Part 2 - Practical Lab: Work on your assignments.** During the second part of the lecture students work on practical assignments. Students work on practical assignments in teams of at most two students. Every practical session students are encouraged to ask for feedback with the lecturer, improving their work over several iterations before finally handing it before the deadline, and the assignments are graded.

- **Exam: Prove you gained knowledge.** At the end of the course, during the exam period, students are required to pass a Moodle exam.

1.2.1 Grading

The course grade consists of two parts: the *practical* grade and the *exam* grade. Grades are calculated as follows.

```
grade(practical, exam) = (grade(practical) + grade(exam)) / 2
  if grade(practical) >= 5.5 and grade(exam) >= 5.5

grade(practical, exam) = "no result", otherwise.
```

This means your grade is the average of practical and exam grades, and both practical and exam grades must be at least 5.5 to pass the course.

---

[2]https://vlo.informatica.hva.nl/main/document/document.php?cidReq=OOAD

1.3 Course Schedule

Table 1.3 shows a week-by-week schedule of activities, assignments, exams and deadlines for the course OOAD. The first column *W* indicates the week number, which enables you to check in your roster what a lecture is about. The second column *Subject* is a brief description of the subjects for that week, and the third column *Homework Chapters* specifies which chapters are self-study reading. Finally, the fourth and fifth column specify which *Practical Assignment* to work on during that week, and which *Exams and Deadlines*[3] are due in that week. Lectures 1-3 focus on Analysis and lectures 4-6 focus on Design.

W	Subject	Homework Chapters	Practical Assignment	Exams and Deadlines
1	Software Methodology, Agile & Requirements.	1, 2, 3, 4	1	
2	Use Cases, Supplementary Specification.	5, 6, 7, 8, 30	1	
3	Domain Model, System Sequence Diagram, Operation Contracts	9, 10, 11, 31	2	Deadline Assignment 1 at lecture start
4	Architecture	12, 13, 14, 38	2	
5	Interaction Diagram, Class Diagram	15, 16	3	Deadline Assignment 2 at lecture start
6	GRASP, Examples GRASP, Visibility, Mapping Design to Code	17, 18, 19, 20, 21, 22	3	
7	Finish the Practical Assignments		3	Deadline Assignment 3 at lecture start
	Exam	All course material (slides, book chapters).		Moodle (50%)

Table 1: Course Plan

---

[3]**Note:** Deadlines are at the beginning of a lecture!

2 Practical Assignments.

Students are required to complete three practical assignments for this course. During these assignments you work in groups of two students. When you have completed the assignment you can request your teacher to approve your work by explaining what you did. Ask your teacher how to deliver the solutions of the assignments. Deadlines are at the beginning of the lesson. Table 2 shows when to work on assignments and deadlines to deliver them.

week	Assignment to work on	Deadline
1	Define a case	
2	Assignment 1	
3	Assignment 2	Deadline Assignment 1
4	Assignment 2	
5	Assignment 3	Deadline Assignment 2
6	Assignment 3	
7		Deadline Assignment 3

Table 2: When to work on assignments and deadlines to deliver them.

Practical Grading

The practical course grade consists of three assignments and is calculated as follows.

```
grade(practical) =
  10 + pt(assignment1) + pt(assignment2) + pt(assignment3)
```

The the function $pt(a)$ refers to the number of points you have scored for assignment a. We describe the grading for each assignment.

Define a case

During the course you are going to define and use your own business case. This can be for example a planning system for sports club, an online shop, a game et cetera. During this course we fulfill the analysis and design part, we are not going to build the app itself. At the VLO you can find some examples (Documents/Lesson_1/Example_Cases).

- **Description.** Define a case for building an Android or IOS app. Make a brief textual description (maximum one A4 paper) that describes the business case. How will your system solve a real-world problem?
- **Approval.** Explain your case to your teacher and ask if the case is appropriate (sufficiently complex and well-defined) for completing the assignments.

Assignment 1

In iterative software development developers practice different disciplines to create and improve so called *artifacts*, tangible pieces of work. You are asked to provide the artifacts described in Table 3 for your case, and package them in a report that conforms to the report guidelines shown in Appendix A. More information about how to develop these artifacts can be found in your textbook [Larman(2004)].

Artifact	Explanation
Vision	Describe the case and high-level goals and constraints.
Use-Case Model	Write four use case scenarios of reasonable size in a *fully dressed style* and illustrate them with a use case diagram, e.g. "Log in" does not count as one of the four use case scenarios.
Supplementary Specification	Describe non-functional requirements and consider reports, documentation, packaging, supportability, licensing et cetera.
Glossary	Make a list of terms and their definitions.
UI Wireframes	Make a visual representation of the user interfaces (these are just ideas, not specifications).

Table 3: Artifacts and Descriptions

1.0 Vision (5pt)

Write a brief description of the business case in terms of high level goals and constraints.

1.1 Use-Case Model (15pt)

A Use-Case Model can be created by performing the following steps:

1. Identify the actors.

2. Identify the goals for each primary actor.

3. Define use cases that satisfy user goals (at least four use cases)

4. Make a use case diagram, use a drawing tool or just pencil and paper.

5. Describe the use-cases, use the template of Table 4. Note that the most important fields are underlined, and some fields may not be applicable. The goal is to provide useful information!

1.2 Supplementary Specification (2pt)

Look at page 104 of your textbook [Larman(2004)] for an example which can help you to make the supplementary specification. You can use the FURPS+ categories.

1.3 Glossary (2pt)

For creating a glossary you can create the template of Table 5.

1.4 UI wireframes (2pt)

Make wireframes/Mock-ups for the app. Search on the internet to find a tool which you can use to make the app wireframes, or use Axure (`http://www.axure.com/`), a tool that use can use for free for 30 days.

1.5 Report (4pt)

Package your artifacts in a report that conforms to the guidelines of Appendix A.

Checklist and grading

When all the artifacts are created, discuss them with your partner and ask if he or she is satisfied or needs to some changes. You can earn 30 points for Assignment 1 according to the following criteria.

- **3pt.** Does the vision statement clearly define the business case?

Use-Case Section	Description of what to fill in this field.
Use Case Name	A numbered brief description that start with a verb, e.g. "U3: buy pencils".
Scope	The system under design.
Level	"user-goal" or "sub-function"
Primary Actor	The primary actor calls on the system to deliver its services. It can be a person or system fulfilling a specific *role*, e.g. "Student".
Stakeholders and Interests	Who cares about this use case, and what do they want?
Preconditions	What must be true on start, and worth telling the reader?
Success Guarantee (postconditions)	What must be true on successful completion, and worth telling the reader?
Main Success Scenario	A typical, unconditional happy path scenario of success. This path consists of a sequence of *numbered steps*.
Extensions	Alternate scenarios of success or failure.
Special Requirements	Related non-functional requirements (if applicable you can refer to requirements specified elsewhere).
Technology and Data Variations List	Varying IO methods and data formats (if applicable).
Frequency of occurrence	Influences investigation, testing, and timing of implementation.
Miscellaneous	Such as open issues (if applicable).

Table 4: Fully dressed use-case template with field descriptions

Term/Concept	Definition
...	...

Table 5: Glossary Template

- **2pt.** Does the vision statement consider goals and constraints from the point-of view of clients/customers and end-users?

- **12pt.** Does the Use Case Model use the fully dressed style use case templates, and does the provided information for each of the four scenarios make sense?

- **3pt.** Is the Use Case Diagram correct UML and does it coincide with the use case scenarios and illustrate it?

- **2pt.** Are appropriate non-functional requirements considered? (e.g., using the FURPS+ categories.)

- **2pt.** Is the glossary a complete list defining concepts necessary for unambiguous communication about the case?

- **2pt.** Can the UI Wireframes be used to explain the use case scenarios?

- **4pt.** Does the report follow the report guidelines of Appendix A?

Assignment 2

In Assignment 2 you continue iteratively developing a system for our case by adding new artifacts, and improving and refining existing artifacts. You are asked to provide the artifacts described in Table 6 for your case, and package them in a report that conforms to the report guidelines shown in Appendix A. More information about how to develop these artifacts can be found in your textbook [Larman(2004)].

2.1 Domain Model (12pt)

For completing the Domain model you should perform the following steps:

1. Identify the nouns and nouns phrases in the fully dressed use cases.

2. Decide which nouns are suitable as conceptual classes.

3. Make a model of the conceptual classes. Use a drawing tool of pencil and paper.

4. Add associations between the classes including naming and multiplicity.

5. Add attributes to the conceptual classes.

6. Discuss the model you have made with the client.

Artifact	Explanation
Domain Model	Make a domain model, a visual representation of real-situation conceptual classes (not of software objects).
System Sequence Diagrams	Make System Sequence diagrams to illustrate the input and output events.
Operation Contracts	Make operation contracts to precise describe system behavior.
Physical or Logical Architecture	Make a deployment diagram to illustrate the physical architecture or a package diagram to illustrate the logical architecture.
Use Case Model	Refine the use-case model of assignment 1.
Supplementary Specification	Refine the supplementary specification of assignment 1.
Glossary	Refine the glossary of assignment 1.
UI Wireframes	Refine the UI wireframes of assignment 1.

Table 6: Artifacts and Descriptions

2.2 System Sequence Diagrams (8pt)

Draw a System Sequence diagram for the main success scenario of each use case [4]. **Note:** These diagrams show communication scenarios between actors and the System as a black-box (we cannot see inside the system).

2.3 Operation Contracts (4pt)

For each System Sequence Diagram choose one operation for which you write a contract. **Note:** We are looking for those system operations that have post-conditions with 1) instance creation or deletion, 2) modified attributes, and 3) formed and broken links between instances.

---

[4]Web Sequence Diagrams (https://www.websequencediagrams.com/) allows you to textually model sequence diagrams.

2.4 Logical Architecture or Physical Architecture (2pt)

Create a deployment diagram to illustrate the physical architecture. Alternatively, create a package diagram to illustrate the layered logical architecture. Use a UML tool for this job.

2.5 Refine the artifacts of Assignment 1

With the new information of the artifacts of the Domain model, System Sequence diagrams and operation contracts you can refine the artifacts of assignment 1.

2.6 Report (4pt)

Package your artifacts in a report that conforms to the guidelines of Appendix A.

Checklist and grading

When all the artifacts are created, discuss them your partner and ask if he or she is satisfied or needs to some changes. You can earn 30 points for Assignment 2 according to the following criteria.

- **12pt.** Is the domain class diagram readable UML, logical and well-defined? Is the diagram well-explained in accompanying text? (-5pt if missing text). Are classes conceptual? (-5pt if software classes).

- **8pt.** Are the system sequence diagram correct UML, and do they describe realistic use case scenarios? (2pt per system sequence diagram). Do they coincide with scenarios in the fully dressed templates? (-1pt if not). Are they accompanied by descriptive texts? (-2pt missing text).

- **2pt.** Does the deployment diagram illustrate the physical context in which the system will run? Is the diagram accompanied by explanatory text? (-2 if missing).

- **4pt.** Do the operation contracts describe operational constraints? (1pt per operation contract).

- **4pt.** Does the report follow the report guidelines of Appendix A?

Assignment 3

In Assignment 3 you continue iteratively developing a system for our case by adding design artifacts. These design artifacts closely resemble object oriented code of the final system, which is outside the scope of this course. You are asked to provide the artifacts described in Table 7 for your case, and package them in a report that conforms to the report guidelines shown in Appendix A. More information about how to develop these artifacts can be found in your textbook [Larman(2004)]. For the design modeling keep the GRASP principles in mind.

Artifact	Explanation
Domain Model	Improve and refine the domain model, a visual representation of real-situation conceptual classes (not of software objects).
Design Sequence Diagrams	Make Sequence diagrams to Illustrate the interaction between software objects (dynamic software behavior / run-time).
Design Class Diagram	Make a Design Class Diagram, a visual representation of software objects (static software structure).

Table 7: Artifacts and Descriptions

3.1 GRASP (4pt)

Apply the GRASP principles in the Sequence Diagrams and the Design Class Diagram. Describe how you have applied the GRASP principles.

3.2 Design Sequence Diagrams (8pt)

Create a design-level sequence diagram for each operation contract you have defined in assignment 2. Use a UML tool (for example ArgoUML[5], UMLet[6] or Web Sequence Diagrams[7]) to model four design sequence diagrams. (2pt per diagram).

---

[5]http://argouml.tigris.org

[6]http://www.umlet.com

[7]https://www.websequencediagrams.com/

3.3 Design Class Diagram (14pt)

Use a UML tool (for example ArgoUML or UMLet) to create the Design Class Diagram of the Design Model. The design class diagram should include:

- Classes, associations and attributes
- Methods (you can mention you omit getters and setters for readability!)
- Attribute type information
- Navigability

Checklist and grading

When all the artifacts are created, discuss them with your partner and ask if he or she is satisfied or needs to some changes. You can earn 30 points for Assignment 3 according to the following criteria.

- **4pt.** Are the GRASP principles applied convincingly? Are the GRASP principles used to make the **design decisions** explicit?
- **12pt.** Is the Design Class Diagram readable UML, logical and well-defined? Is the diagram well-explained in accompanying text? (-5pt if missing text). Are the associations directed?
- **10pt.** Are the sequence diagrams correct UML, and do they describe realistic run-time scenarios? (2.5pt per system sequence diagram). Do they coincide with methods in the class diagrams (-1pt per method if not, but getters and setters may be omitted if stated that they are left out intentionally). Are they accompanied by descriptive texts? (-2pt missing text).
- **4pt.** Does the report follow the report guidelines of Appendix A?

A Report Guidelines

Here we provide general report guidelines that help you to structure a deliverable. Please note that this manual conforms to these report guidelines.

Cover

A report cover shall have a descriptive title, authors, a version number and a date, e.g. as on page 1 of this manual.

Changes

You can track changes different authors made to different versions document over time. We have omitted change tracking in this manual, but an example of how to track changes in a table is shown in Table 8. Please let your teacher know which author added what artifact in your report.

version	change	author
0.01	Modified last years assignments and added grading.	Riemer van Rozen
0.01	Added a report guideline in Appendix A.	Riemer van Rozen
0.02	Rewrote the introduction.	Riemer van Rozen
0.03	Removed interview reference.	Riemer van Rozen
0.04	Fixed the fully dressed use case template.	Riemer van Rozen
0.05	Fixed difference between physical and logical architecture	Riemer van Rozen
0.06	Fixed GRASP spelling	Riemer van Rozen
0.07	Fixed Assignment 3 Checklist & Grading	Riemer van Rozen
0.07	Added URLs of UML tools	Riemer van Rozen
1.00	Finalized the manual and released it.	Riemer van Rozen
...	...	...

Table 8: Tracking changes in a table

place holder

Figure 1: How to refer to a figure from text

Introduction

The introduction shall describe what the document is about, explain in a few sentences what its contents are and who is the intended audience (reader) of the document, e.g., some documents are technical whereas others are more business-oriented.

Report Contents

Several sections (or chapters) of the document describe the actual content of the report. For our assignment it makes sense to use one section per artifact. We never provide a figure, model, image or a table without a caption, a numbered description appearing below it. There shall always be an accompanying text referring to the number that explains what it is supposed to explain to the reader. If a figure does not explain anything it is redundant! Here follows a reference to an example table. Figure 1 explains how to refer to a figure from text.

References

[Larman(2004)] Craig Larman. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition)*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2004. ISBN 0-13-148906-2.