Computer Organization 2024
Course Guide

Authors: Alexandru Iosup[1], Tiziano De Matteis[1], Animesh Trivedi[1],
Radu Nicolae[1], Lennart Schulz[1], Dovydas Vadišius[1]
With contributions from:
Paul Ellsiepen[1], David Breitling[1], Alexandru Uţă, Tim Hegeman[1,2], Stefan Hugtenburg[2], Otto W. Visser[2]

[1] VU University Amsterdam, [2] TU Delft

1 Introduction

Computers are everywhere, in academia, governance, industry, and many other activities that impact our society. But what are computers? How do they work? How to analyze them and improve their performance? Matching the requirements of the ACM/IEEE CS Curricula 2023, Computer Organization 2024 includes: the architecture, the structure, the operation, and the interconnection of computer components into computer systems, including modern architecture, data representation, assembler programming, virtual machines, structure of translators, compiling, and loading, basic operating system concepts (e.g., I/O, interrupt handling process).

The complete information about the course structure, assignments, deadlines, communication, et cetera is on Canvas. This document comes in completion to the information provided via the Canvas page and presents our teaching philosophy, course principles, and policies. This document is structured as follows:

Section 2 discusses gamification and its concepts. Section 3 presents how the course aligns with the IEEE/ACM and the VU curricula. Section 4 explains the structure of the course. Section 5 introduces part of the material available for the course. Section 6 discusses the assistance we offer and the grading procedure for key components of the course. Section 7 states the requirements for excelling and for completing this course. Section 8 discusses how to participate in the course, how to sign up for various activities, and how to communicate with the teaching team. Section 9 explains the policy of our deadlines. Section 10 summarizes grading. Section 11 explains the zero-tolerance policies we have against academic fraud and discrimination. Appendix A presents three frequently asked questions. Appendix B shows multiple examples of student profiles and their chosen paths through the course.

2 Gamification

The BSc Computer Organization course uses gamification. In our course design, the core of gamification is to stimulate students to take steps, towards their own chosen path, in their course journey without fear of failure. This course rewards students with three types of points:

1. **Points** - offered for achieving mandatory milestones of the course (announced at the beginning of the course).
2. **Extra Points** - offered for achieving milestones in the tasks chosen in the study path (e.g., self-study, quizzes, lab extra assignments) (announced at the beginning of the course)
3. **Bonus Points** - reward for exceptional, excellent academic behaviours (not announced at the beginning of the course)

Gamification helps with accommodating many types of student abilities and desires. However, “with great power comes great responsibility”. Our approach to social gamification allows students to choose their own path of advancement:

- Students can attend the lecture(s), tutorial(s), and lab session(s) they want.
- Students can choose which type of activity to engage with. From being active in lectures and tutorials to programming and demonstrating their technical and teamwork skills in the labs, to working in a team and solving exercises in the Self-Study activity.
- Students can choose the desired difficulty of exercises, beyond the basic requirements.
- Students can choose how to construct their own grade. They will combine large rewards from exams and the chosen lab exercises, and small rewards from in-class activity and quizzes.
- Students choose to expand their knowledge for the sake of knowledge. Outside the lecture courses, the best students, with exceptional grades, will get invited to extra lectures.
- Alternatively, students can also choose the classical path: do well in the exam and in the lab. All good, if it works for you.

Students can get the most out of this course if they actively try to find the component they enjoy most and excel in it. In particular, students need to put into this course the *6 ECs = 168 hours* of study time. Only answering correctly 50% of the exam questions is not enough.

Other courses that employ gamification in the curriculum at VU Amsterdam are Computer Networks, in the BSc, and Distributed Systems, in the MSc. Other courses that employ gamification in the Computer Science curricula in the Netherlands include the BSc course on Computer Organization at TU Delft.

3 Alignment with IEEE/ACM CS Curricula 2023 and CS Curriculum at VU Amsterdam

This course follows the IEEE/ACM Computer Science Curricula 2023 (final version), covering the Body of Knowledge Architecture and Organization, specifically:

- AR1. Digital Logic and Data Representation [core]
- AR2. Computer Architecture and Organization [core]
- AR3. Interfacing and I/O Strategies [core]
- AR4. Memory Architecture [core]
- AR5. Functional Organization [core]
- AR6. Multiprocessing [core]
- AR7. Performance Enhancements [elective]
- AR8. Directions in Computing [elective]

This course sets up the basic knowledge, for all the Computer Systems courses in the VU Computer Science curriculum:

- BSc Computer Networks
- BSc Operating Systems
- MSc Distributed Systems
- MSc Parallel Computing

- MSc Systems Security

This course complements other basic courses on:

- Programming Fundamentals
- Reasoning and Logic
- (Advanced) Data Structures and Algorithms
- Software Engineering
- Cloud Computing
- System Design
- Peer-to-peer systems

4 Course Structure

The BSc Computer Organization course teaches the basics of digital computers (general requirement). It is a complex first-year course, that covers a variety of topics: the architecture, the structure, the operation, and the interconnection of computer components into digital computer systems, including modern architectures, data representation, assembler programming, virtual machines, the structure of translators, compiling and loading, basic operating systems concepts (I/O, interrupt handling, process), parallelism and synchronization, et cetera. A bit of everything to open students' appetite for more.

This course includes various activities, which are theoretical, practical, or both

1. *[THEORY]* **In-class participation**, during which students can answer questions asked by the lecturer, solve exercises, and participate in quizzes.
2. *[THEORY & PRACTICE]* **Tutorial**, during which students learn how to solve representative, yet basic problems, linking the theoretical and the practical components of the course.
3. *[THEORY & PRACTICE]* **Self-Study**, during which students will solve textbook exercises, within groups of up to 6 students.
4. *[PRACTICE]* **Lab**, during which students will apply the theory in a practical context, by solving problems of various scopes and difficulty, alone or in a team of 2 students.
5. *[THEORY & PRACTICE]* **Extra lectures** (for selected students), during which students learn about state-of-the-art topics by engaging with the theory and practice of the field. We will cover several such fields, subject to the performance of this year's group.

If students excel in any of the in-class participation, tutorial, self-study, or lab, they will be invited to take part in extra lectures, which cover more advanced topics. Extra lectures do not count towards the final grade but teach about the state-of-the-art "hot" topics in Computer Science. For each of these topics, we have an interactive lecture covering the basics and a large session of hands-on experience. Have you ever wanted to drive ±1,000 core GPUs with one program? How about commanding a datacenter with ±1,000 core CPUs? Then the extra lectures are for you!

5 Material for the Course

This course offers a diverse set of materials for student's preparation. We offer two textbooks to choose from, lecture slides, academic literature, self-study exercises, and a lab guide.

[IMPORTANT]: We expect students to prepare well before every session, following the Reading Guide. This implies reading the related chapter from the chosen book and the academic literature given.

5.1 Textbook

This course distances itself from the use of a single textbook in the educational process (i.e., reading and studying material). We base the course and recommend students to choose and read *any* of the following textbooks:

1. Carl Hamacher and Zvonko Vranesic, Computer Organization and Embedded Systems, 6th edition, McGraw-Hill Education, 2011. ISBN-13: 978-0073380650
2. David A. Patterson and John L. Hennessy, Computer Organization and Design: The Hardware/Software Interface, 5th edition, Morgan Kaufmann, 2013. ISBN-13: 978-0124077263

5.2 Lecture and Tutorial Material

This course utilizes various slide decks for the Lecture and Tutorial sessions. All the material is shared on Canvas *after the session*. This creates a desirable challenge in taking notes and summarizing the material presented live.

The material presented in class is extending, complementing, and correcting the material present in the textbooks; it is often up-to-date, whereas the books have been printed in years past. Hence, when in doubt, the material from the slides takes precedence over the material in the textbook. The lecture and tutorial material is also exam material.

5.3 Self-Study Material

The material for the Self-Study consists of exercises at the end of the recommended chapters in any of the textbooks. The precise list of chapters and exercises can be found in the Self-Study overview.

5.4 Lab Material

The material for the Lab consists of a Lab Guide, which offers:

- Guidance over x86-64 Assembly, used in the Lab component,
- Mandatory lab exercises, and
- Extra exercises.

The number of points is directly proportional to the difficulty of the exercise, students can choose the desired difficulty and exercises.

6 Grading Process and Assistance

There is one coordinator for each activity: Lectures, Lab, Self-Study, Tutorial, and Exams. See the Canvas Teaching Team page for the coordinator(s) of each activity. Several Teaching Assistants (aka TAs) will help with the Lectures, Labs, Self-Study, Tutorials, and Student-Teaching Team communication. The task of the coordinators is to help with *extreme* cases, in particular, cases that cannot be resolved fairly by TAs. The tasks of the TAs are to help with *difficult* cases, to *verify* students' results, and *to grade* the extra assignments. TAs do not help in solving the exercises, this is the students' task.

6.1 Self-Study

No sessions are scheduled for group meetings. We expect students to organize periodic meetings with their groups. For example, students can meet before, or after the Tutorial or Lecture sessions. The grading process includes intermediary milestones and concludes at the end of the course by considering all their self-study work. We assess the correctness of the exercises and the submission quality.

We highly encourage using our LaTeX template.

6.2 Lab

We reserved three sessions, each a block of 3h 45min, where students can demonstrate their practical skills. We expect students to join the sessions with their teammates and hand in the assignment to the TA. The reward for the practical work is offered only by handing in the assignment to the TA, who will ask questions and ensure the authenticity and correctness of the solution. The grading process for the labs includes intermediary milestones and concludes at the end of the course by summing up all the points earned via lab work. Extra and bonus points will also be considered.

7 Requirements for Excelling and for Completing this Course

To complete this course with a grade of 10 (maximum), students must accumulate at least *9,750 points*. To pass the course, students need to accumulate at least *5,500 points*, meet the *exam passing criteria*, and pass all the mandatory lab exercises. If a student doesn't meet the *exam passing criteria*, or the mandatory lab exercises criteria, the final grade will be capped at 5,000 points.

Exam passing criteria

Students need to pass at least 25 questions in total for the midterm (the midterm component has a total of 20 questions) and for the final (the final component has a total of 30 questions). The highest grade is taken into account for both the midterm and the final. The answer to "do I meet the exam passing criteria?" can be given also via the following

```
# m1 means attempt to take the midterm exam, on Feb 29
# m2 means attempt to take the midterm exam, on Mar 27
# m3 means attempt to take the midterm exam, on Jun 5
# f1 means attempt to take the final exam, on Mar 27
# f2 means attempt to take the final exam, on Jun 5
# x1 means the maximum number of correct questions across the midterm components
# x1 means the final number of correct questions across the final components

def didPass(m1, m2, m3, f1, f2):
   x1 = max(m1, m2, m3)
   x2 = max(f1, f2)
   return (x1 + x2) >= 25

# Example 1: Student participated in the midterm exam on Feb 29, correctly answering 12/20
    questions; participated in the midterm on Mar 27, result 3/20; participated in the final
    exam on Mar 27, result 14/30; and skipped the re-sit on Jun 5 altogether.
# input: didPass(12, 3, 0, 14, 0) => output: return (12 + 14 >= 25) => pass

# Example 2: Student did not participate in any midterm exams, but correctly participated in
    the final exam on Mar 27, result 25/30; and skipped the re-sit on Jun 5 altogether.
# input: didPass(0, 0, 0, 25, 0) => output: return (0 + 25 >= 25) => pass
```

```
# Example 3: Student participated only in the midterm exam on Feb 29, correctly answering 20/20
    questions; skipped the midterm exam and final parts on Mar 27 and re-sit on Jun 5
    altogether.
# input: didPass(20, 0, 0, 0, 0) => output: return (20 + 0 >= 25) => fail
```

To understand how to accumulate points, see also Section 10.

8 Participating, Signing Up, Communication

8.1 Participating

We encourage all the students to actively participate. The physical presence may also be a form of participation, but not the one we are looking for. Active participation could take the form of answering questions, asking questions that are also relevant for others, engaging successfully in quizzes, showing practical skills in the labs, and demonstrating deep knowledge about course topics by solving exercises with the team.

We expect students to engage with the material also at home, individually and with their team. Be fair about it, do not free-ride, and make and then fulfill clear agreements with the team about who does what and when.

8.2 Signing Up

Every student can participate in any main course activity. Yet, students need to sign up:

- **The course itself.** Students signed up for the course by registering through the VU's registration module. We advise students to sign up only if they are aware and agree with the requirements of this course, especially in terms of the workload of the course (168 hours spread over 2 months, the equivalent of 6 ECs).
- **Any team-activity (e.g., Self-Study, Lab)**. By signing up with the team, students hereby indicate they are aware of and agree with the social contract of helping their team succeed. In case a student wants to join a team, but does not have one, or a team seeks more members and cannot find any, we advise students to let the Teaching Assistants (TAs) know about this and request help.
- **Self-Study**, Students who want to participate in the Self-Study components should form a group via Canvas -> People. For these activities, students can work in *teams up to 6 students*. Teams with fewer people, even 1, are accepted, yet the required set of exercises will not be changed.
- **Lab** Note this is a *mandatory component* to pass the course. Students must enrol for the Lab via Canvas, by selecting a time slot they can join. *Attendance of the Lab sessions is not mandatory*. It is required to have all the mandatory exercises verified by a TA. Students who register for a lab session time slot can also attend labs from other time slots but will have *decreased priority* in the process of answering questions and reviewing assignments. For the lab exercises, students can work in *teams up to 2 students*. Teams of 1 person are accepted, but the workload will not be decreased.

For the extra lectures, students are selected based on their performance in the course, in any of the main activities.

8.3 Communication Guidelines

The teaching team can be contacted via one of the following methods:

1. **Office Hours**. On Thursdays, 13:30-14:30, we offer weekly office hours, during the course period, where we will answer questions. We meet via Zoom. The meetings are organised between W6 - W13 (weeks of the year).
2. **Unplanned questions**. Students can ask questions via the following Google Form.

We answer messages received only via one of the methods above. We do not answer messages received through other communication methods.

Please also restrict questions to those where the answer is not already present (e.g., in the lab manual, on Canvas, in an announcement on Canvas).

8.4 Asking for Exceptions to the Course and University Rules

Sometimes, a special situation warrants giving a delay, offering one more examination, or other exceptions to the course and university rules. *We can only make exceptions via the academic advisor.* In case of personal problems, please contact them.

No points can be transferred from previous Computer Organization editions.

9 Deadline Policy

This course has *strict deadlines*, which cannot be extended. The detailed lists of deadlines can be found on Canvas.

10 Grading and Points

We expect students to focus more on learning about computer organization and on thinking critically about the topics of this course, and less on maximizing the earned points. The points come naturally with the gained knowledge. The grading philosophy of this course reflects this expectation: students receive points not only for regular (pre-defined) activities, but also for exceptional, additional work. At the end of the course, the points obtained by a student across all course activities are summed to compute the student's total score. A student's overall grade for the course is derived by dividing their total score by 1,000. The resulting value is rounded as the nearest multiple of 0.5 and capped at 10.

- **Exception 1:** Per VU policy, a grade of 5.5 is never awarded, so total scores between 5,500 (inclusive) and 5,750 are awarded with a passing grade of 6 instead. Scores between 5,250 and 5,500 (exclusive) are awarded a failing grade of 5.
- **Exception 2:** If a student did not complete the mandatory lab assignments, or does not meet the exam passing criteria mentioned in Section 7, they cannot pass the course and will have their grade capped at 5.

Points are not transferable between the years. Hence, we do not transfer any points earned in the past years. Points are valid only during the academic year.

10.1 Point distribution

- **Exam:** *The Midterm Exam* has 20 questions, with 10 questions of a baseline difficulty, each worth 20 points, and 10 questions with a normal difficulty, each worth 300 points. *The Final Exam* has 30 questions, with 15 questions of a baseline difficulty, each worth 20 points, and 15 questions with a normal difficulty, each worth 300 points. For both Midterm and Final exams, questions are separated into difficulty groups and the number of points is indicated per each question.
- **In-class:** During lectures, students can earn 50 points for in-class activity, if they answer Menti questions correctly.
- **Self-Study:** The summed-up maximal points for Self-Study is 1,500 points. This is divided into *Self-Study Part 1*, where teams can solve up to 2 chapters, and *Self-Study Part 2*, where teams can solve up to 4 chapters. Each chapter is worth 200 points in terms of correctness and completion, and 50 points in terms of document quality and formatting. Using the provided LaTeX template maximizes students' chances of receiving maximum points for the quality check.

- **Tutorial:** Up to 500 points. Students have the opportunity to tackle a *Golden Exercise* individually, valued at 50 points, both before and after each Tutorial session. There will be 10 such *Golden Exercises*.
- **Lab:** The maximal reward is 4,900 points, for regular and bonus assignments, depending on the quality and depth of students' work. The mandatory assignments (1-4) must be completed before any other Lab bonus is awarded. Completion of all mandatory assignments is awarded *500 points*. Extra assignments are awarded up to *1,000 points* each.
- **Quizzes:** Students can earn up to 1,250 points from in-class quizzes. These quizzes are not announced in advance (i.e., they are not announced before the sessions).
- **Extra opportunities:** We offer students extra opportunities, up to 1,000 points, to earn points *during* the course, which are not announced from the beginning of the course.
- **Exceptional behaviour:** The teaching team can reward students with exceptional academic behaviour in every component of the course.

10.2 Point Summary

The table below summarizes the primary sources of points in the course.

	Source	Maximum Reward
In class activity	Lecture Menti	700
	Quizzes	1,250
	Golden exercises	500
Lab	Lab Mandatory	500
	Lab Extra	4,150
	Miscellaneous	250
Exams	Midterm	3,200
	Final*	4,800 (+3,200)
Self Study	Part 1	500
	Part 2	1,000
Miscellaneous	Extra opportunities	1,000
Total available points**		17,850

Table 1: Point distribution summary

* For the final exam, students can retake the mid-term exam. We will automatically use the maximum score obtained by the student in each component.

** The total of 17,850 is not designed to be achieved within the time limit imposed by the 6 ECs offered for this course. However, we do see a clear point in offering a broader scope and horizon, both for high-achieving students who want to compete and for all other students who want to explore and discover their strengths.

11 Zero-Tolarance Policies

We have two zero-tolerance policies:

1. We are anti-fraud, so we want to prevent all academic fraud that may happen in this course, see Regels en Richtlijnen (Rules and Guidelines), Art 19 & 20.
2. We are anti-discrimination, so we want to prevent all situations when a student or a staff member, or groups thereof, are discriminated against or are subjected to inappropriate conduct.

Appendix A

Is presence at the lectures mandatory?

No, but the discussion during the sessions make them worth attending. Additionally, the bonuses offered during and at the end of the lectures contribute to a higher grade.

Why are there so many course activities? Lectures and Lab and Tutorial and Self-Study and...? Do I need to do all of them?

Firstly, we're glad you asked. Learning about computer systems requires specialized approaches, which are not the same as you would normally find in, say, a course on mathematics or theoretical computer science. Computer systems is a broad field, with much theoretical and practical depth, and thus requires a mix of didactic approaches. Over more than one decade of gamification, we found a combination of activities, that include lectures, solving problems that link theory and practice, hands-on activity, interaction in the classroom, and teamwork. allows most students to develop deep knowledge about the topic.

Secondly, students do not have to pursue all the course activities. But if they want to, they can – but they should mind the time they invest in this! Students can mix and match. See also Section 2 on why we do this, Section 10 for the detailed grading system, Section 11 for examples of how different students chose to approach this course, and a quick summary in the next frequently asked question.

Q: Why is the grading system so complicated? Can I just focus on the exam and complete the mandatory exercises for the lab?

Firstly, we see our grading system as unusual for university courses, rather than complicated. However, similar systems are used throughout the world in technical education, and every year many CS students become more knowledgeable because of them [1, 2].

Secondly, this course intends to allow students some degree of freedom in defining their own path of advancement, while not removing the (internationally recognized) learning objectives related to distributed computing systems. We give various opportunities for exceptional activities (extra-credit and bonus-worthy), see Section 10 for several examples. In plain English, *as a student in our course, you can choose what to put the time in, yet still be rewarded for it.*

Thirdly, it is exactly the second part of the question that should tell students that different people have different ways of approaching education. Students can do exactly what you asked. However, just like you, other students would ask about the possibility of focusing on other activities, which stimulate them more. Offering several, albeit, not too many, ways to address the "Can I just ..." question is what the grading system offers.

Appendix B

To help students identify their path through our course, we present in this section several example profiles of students representing different personas. The table below depicts for each student profile a breakdown of points they might receive per type of activity. A more detailed description of each profile and the kind of student they may apply to can be found below the table.

Disclaimer: the example profiles described in this section should not be interpreted as a guideline or guarantee on how many points students should or will obtain from a specific activity. Instead, the examples serve as illustrations of some of the paths that students may take in this course.

Profile 1: The Traditional Student

This student prefers a traditional approach to education. They focus primarily on and score above average on the exams. They complete the mandatory lab assignments to pass the course and participate in self-study to prepare for

the exam.

Profile 2: The Coder

This student prefers programming over solving exercises on paper. They have experience with (low-level) programming or quickly pick up on new languages and concepts. They spend most of their time on bonus assignments at the lab but also put enough effort into the exams to pass the mandatory questions and a decent total score for the course.

Profile 3: The Perfect Student

This student gets the most out of this course by participating in all activities and putting in the effort to get high scores across the board but staying within the time limits (6 ECs).

Profile 4: The Basic All-Rounder

This student does not excel at written exams but compensates by participating in many activities to gain points and knowledge. They get average scores in each activity to end up with a passing grade. However, they miss out on a lot of points by not finding a type of activity they can do well at. They also risk failing the course if they perform worse than expected in any activity.

Profile 5 (not in the table): The Do It All

Doing everything to the maximum allowed by design is unlikely, time-wise. We strongly advise you to not try to do this and focus on The Perfect Student instead.

	Source	Maximum Reward	Traditional	Coder	Perfect	Basic All
In class activity	Lecture Menti	700	350	0	600	350
	Quizzes	1,250	250	250	1,000	450
	Golden Exercises	500	100	0	400	100
Lab	Lab Mandatory	500	500	500	500	500
	Lab Extra	4,150	0	4,000	3,000	0
	Miscellaneous	250	0	250	250	0
Exams	Midterm	3,200	1,980	1,000	2,600	1,360
	Final	4,800	2,700	1,600	3,600	2,340
Self Study	Part 1	500	500	500	475	275
	Part 2	1,000	500	1,000	950	450
Miscellaneous	Extra Opportunities	1,000	350	200	850	350
Total		17,850	7,230	7,800	14,225	6,175

Table 2: 4 possible student profiles

References

[1] A. Iosup and D. Epema. On the gamification of a graduate course on cloud computing. In *The International Conference for High Performance Computing, Networking, Storage and Analysis. IEEE*, 2013.

[2] A. Iosup and D. Epema. An experience report on using gamification in technical higher education. In *Proceedings of the 45th ACM technical symposium on Computer science education*, pages 27–32, 2014.