Organization

This unit contains more information about the organization of the course. Please read it carefully before the first lecture, so that you are familiar with the most important aspects!

Prerequisites (overview)

You passed Praktikum: Grundlagen der Programmierung (IN0002) or a similar course
You have experience in object oriented programming with Java
You have your own computer and use it during the lectures and exercises!
Detailed prerequisites

You need detailed experiences in object oriented programming to participate successfully in this course. You should be able to …

Understand Java syntax, create variables, write conditional statements (e.g. if and switch) and control structures (e.g. traditional for loop, enhanced for loop and while loop)
Use Java's data types (e.g. bool, int, long, float, double, String) and operators (e.g. =, ==, !=, <, >)
Create and invoke static and non-static methods
Understand how to use collection types (e.g. Array, List, Set, Map, Stack) with generics
Understand the differences between compile time and runtime, and between compile time types and runtime types
Understand the differences between classes and objects
Create classes, instantiate objects, use attributes, methods, constructors and associations
Use modifiers to apply information hiding
Apply the concepts of encapsulation, polymorphism and inheritance
Write abstract classes and interfaces
Understand the differences between overriding and overloading
Apply exception handling
Use lambda expressions and streams
Create simple graphical user interfaces with JavaFX
Use Maven (pom.xml) or Gradle (build.gradle) to configure dependencies
Write simple test cases with JUnit 5
Interact with git repositories (clone, commit, push)
Online material to refresh your knowledge / prerequisites

Interactive online tutorial: https://www.learnjavaonline.org
Java beginners tutorial: https://javabeginnerstutorial.com/core-java-tutorial
Core Java tutorial: https://www.studytonight.com/java
In German: Online-Kurs Lernen objekt-orientierter Programmierung https://www.edx.org/course/lernen-objekt-orientierter-programmierung
In German: Java ist auch eine Insel http://openbook.rheinwerk-verlag.de/javainsel
Learning goals of the course (high level)

You understand the concepts and methods of the different phases of a software engineering project
Examples: model the problem in a UML diagram, reuse classes and components, deliver the software to the end user
You can select and apply suitable concepts and methods for concrete problems in complex and large projects
You can apply agile methods in teams

You understand the most important software engineering terms and workflows You can analyze, evaluate and solve concrete problems in software engineering, e.g. with the help of design patterns
Learning goals of the course (detailed)

After this course, you are able to ...

Model abstractions of the software system with different kinds of UML diagrams
Understand process models in software development: linear, iterative, agile (Scrum)
Apply requirements elicitation and analysis techniques
Create a functional, a dynamic, and an object model
Apply system design principles such as specification of the software architecture, usage of architectural patterns, and prioritization of design goals
Apply object design principles such as reuse, design patterns, and interface specification
Understand the differences between component, integration, and system test
Write test cases using JUnit and apply testing patterns such as the mock object pattern
Apply software configuration management, continuous integration, and continuous delivery
Understand the concepts of software quality and apply static code analysis
Understand software maintenance and evolution
Apply modern project organization and communication methods
Interactive learning

Goal: directly involve students in the learning process using digital tools
Engage students in two aspects – doing things and thinking about those things
Multiple lecture units per week
We teach you one (or more) concepts
You apply these concepts in lecture exercises
You think about the concepts you are applying
Artemis supports interactive learning
Tutor groups
Group exercises
Homework presentation
Team project
Lecture

The lecture is divided in three main parts, each consisting of smaller units. If you participate in the lecture hall (Garching, Galileo Audimax), please come early to get a seat
12:15 - 13:00
13:05 - 13:50
16:15 - 17:00
There will be a livestream on https://artemis.in.tum.de/courses/169/lectures and https://live.rbg.tum.de
Lecture recordings will be available after a few days (due to editing) in small videos corresponding to the lecture parts (typically 3-6)
Thanks to Ruth Demmel, Andreas Jung and the whole team of the RBG for their support!
Tutors will be available, assist you in lecture exercises and answer questions
We will distribute the lecture slides in parts right before the lecture (or the corresponding part) starts
You can combine all lecture parts into one PDF on Artemis
Copyright

You are not allowed to publish them on the internet
Textbook

Bernd Bruegge, Allen H. Dutoit, Object-Oriented Software Engineering: Using UML, Patterns and Java, 3rd Edition, Publisher: Prentice Hall, Upper Saddle River, NJ, 2009; ISBN-10: 0136061257, ISBN-13: 978-0136061250

30 copies of the English version available for free for TUM students on https://ebookcentral.proquest.com/lib/munchentech/detail.action?docID=5832631
Login with your TUM Online account
Read chapters of the book online (some of them can be downloaded as PDF)
German version available as PDF for free on https://artemis.in.tum.de/courses/169/lectures/323
Lecture exercises

Intertwined with each lecture, usually after some content has been presented, or to clarify some concepts
Published during the lecture, to be solved until the end of the day
Prepare for homework exercises and for the final exam
Quiz: at the beginning of each lecture about material from the previous lectures: always starts at 12:05 pm (CEST) with a ~5 min duration
Tutor groups

70 tutors groups on Wednesday, Thursday and Friday
64 onsite groups and 6 virtual tutor groups
No obligation to attend, but highly recommended
In case your tutor group meeting is cancelled (SVV, FVV, holiday): reallocate yourself to another tutor group
Typical schedule of tutor group meetings

Short review (max. 5 min) of the lecture content (no repetition)
Your tutor quickly shows the roadmap, objectives and summary slides of the lecture
You can ask questions, if something is unclear (prepare them in advance)
Presentation and discussion of previous homework (30 - 45 min)
Present your solution to your tutor and the other students
Lecture recap quiz (5 min)
Participate in a small quiz (exclusively in the tutor group)
Discuss the quiz results afterwards shortly
Group work (30 - 45 min)
Solve small tasks in the group and present your solution
Short discussion of the new homework (max. 5 min)
Your tutor quickly shows the new homework
You can ask questions, if something is unclear (prepare them in advance)
Registration for tutor groups

Please enter your preferences on TUMonline between Tuesday (April 26) and Thursday (April 28)
Choose your preferred tutor groups: List of all Tutor Groups
Select groups based on their language (EN / DE)
Select groups based on their location (INNENSTADT / GARCHING / VIRTUAL)
Select groups based on their time slots
Please select as many as possible! Otherwise you might not get matched!
🔗 TUMonline registration procedure

Receive your assignment on Friday (April 29) on TUMonline
The first tutor group meetings take place on Wednesday (May 4)
Exception handling: if you missed the matching, could not be matched or want to change your group
Late registration or a change of the tutor group in TUMonline after the matching results have been published
Participation in virtual tutor groups

Check the slides "Tools: Communication & Tutor Groups" on https://artemis.in.tum.de/courses/169/lectures how to participate
Quick summary:
Make sure you are registered in a virtual tutor group (check TUMonline)
Find the link for your weekly meeting in TUM-Conf (Zoom)
Join the Zulip stream of the tutor group, e.g. #EIST22-Wed23 ➡️ the Zoom link should be posted in the channel
View the Zoom link on https://artemis.in.tum.de/courses/169/lectures
Click on the Zoom link next to your tutor group, e.g. Mi 01
Join with the Zoom desktop app or with the browser (a Chromium based browser is recommended)
Code of Conduct

Please read the 3 documents on https://artemis.in.tum.de/courses/169/lectures/324 carefully and make sure to respect them!
Violations may result in a permanent exclusion from attending tutor groups, interacting in Zulip, or even participating in the exercises!
EIST student code of conduct (summary)

Respect others and their opinions
Follow the laws
Use your real identity
Stay on topic
Participate actively
Wait for responses
Do not post solutions
Do not record tutor groups
Choose an appropriate background
Have a good time 🙂
Exercises

There are 4 main exercise types in EIST

Lecture exercises (see above)
Group exercises
Solve a task in a small group during the tutor group meeting
Prepare for homework exercises and for the exam
Homework exercises
Published after the lecture: to be solved within a week at home
Assessed automatically (real time feedback) or manually by the tutors
(individual feedback after one week)
Presented and discussed in the tutor groups
Team project
Experience agile software engineering in a small team with a continuous project context

Done as homework
Homework

Published every Tuesday evening on Artemis
You have 1 week to solve the homework on Artemis
You can only submit solutions to exercises on Artemis
Each exercise has a due date
If you miss the deadline, your solution will not be assessed
You present your solution after the due date in your tutor group and discuss the solution with other students
Note: every student has to solve and submit the homework individually (group work is not allowed in homework exercises, we will check for plagiarism)
There are no exceptions and no warnings
Cheating ("Unterschleif") leads to exclusion from the exercises
Tutors will assess your solution and provide the score and the individual feedback after 1 week on Artemis
Homework assessment and complaints

Artemis uses double blind assessments to improve the fairness
The tutor does not know the identity of the student
The student does not know the identity of the tutor
If you think the assessment is wrong, you can complain on Artemis
After you have received the result on Artemis, you have 7 days to complain
You have 3 open complaints for the whole semester
Each complaint will be reviewed by a second tutor
If your complaint is accepted, your complaint count stays the same
If your complaint is rejected, your complaint count decreases by 1
Alternative: if you do not fully understand the assessment, you can ask for more feedback
More feedback requests cannot change the assessment, tutors will only add more explanations
Note: in case you are not satisfied with the assessment, you have to decide: either complain or ask for more feedback (both is not possible)
Team Project

Experience a small agile software engineering project in a team of five students
Choose one of three predefined problem statements or come up with your own problem statement
Work on the chosen problem statement together with your coach (tutor) in a continuous context
Develop a REST based client-server application using Spring Boot with a client framework of your choice (e.g. JavaFX, Angular, React)
Possibilities of customization: incorporate your own creative ideas
Schedule

Sprint	Start	End	Duration	Main purpose
	Mon 02.05.22	Fri 13.05.22	2 weeks	Team forming
1	Mon 16.05.22	Fri 27.05.22	2 weeks	Development
2	Mon 30.05.22	Fri 10.06.22	2 weeks	Development
3	Mon 13.06.22	Fri 24.06.22	2 weeks	Development
4	Mon 27.06.22	Fri 08.07.22	2 weeks	Development
5	Mon 11.07.22	Fri 22.07.22	2 weeks	Development
	Mon 25.07.22	Fri 29.07.22	10 min	Final presentation

We will announce more details later

Exam

Final and repeat exam will be online exams on Artemis (in its exam mode), also called graded online exercise (GOE). You can participate from home.

Date: Monday, 08.08.2022, 17:15 - 18:45 (preliminary scheduled)
Time: 90 min, Points: 100
Location: online on Artemis (exam mode), open book, no communication allowed
Focus: problem solving, application of knowledge, similar to exercises (learning by heart will not help)
Hint: participate in the exercises and in the team project to prepare best for the exam
Bonus: up to 10 exam points by successfully completing exercises (more details, see below)
Requirement: present 2 homework exercises in the tutor group
Note: The repeat exam is preliminary scheduled on Monday, 12.10.2021, 11:00 - 12:30. Important: The bonus is not applicable.

Typical exam structure

~ 20 % quiz questions
~ 15 % modeling exercises
~ 15 % text exercises
~ 50 % programming exercises
The exam focuses on problem solving and the application of software engineering knowledge. Exam questions will be similar to lecture and homework exercises. Learning by heart won't help you!

Preliminary grade key

GOE points	➡️	GOE grade
[0, 35)	➡️	5.0
[40, 45)	➡️	4.7
[45, 50)	➡️	4.3
[50, 55)	➡️	4.0
[55, 60)	➡️	3.7
[60, 65)	➡️	3.3
[65, 70)	➡️	3.0
[70, 75)	➡️	2.7
[75, 80)	➡️	2.3
[80, 85)	➡️	2.0
[85, 90)	➡️	1.7
[90, 95)	➡️	1.3
[95, 100]	➡️	1.0

$(a, b] = \{ x \mid a < x \leq b \}$ $[a, b) = \{ x \mid a \leq x < b \}$

Bonus system

Prerequisite: you have passed the GOE
Additional requirement: you must present the solution to at least 2 homework exercises in your tutor group
Important: the bonus system is not applicable to the repeat exam
Participate in exercises to receive points

Lecture: 10 points per lecture (total ~ 120 points)
Homework: 20 points per week (total ~ 220 points)
Team work: ~ 200 points
Mapping of exercise score to additional GOE points

Exercise score ➡️	Additional GOE points
[0%, 5%) ➡️	0.0
[5%, 10%) ➡️	0.5
[10%, 15%) ➡️	1.0
… … …	
[90%, 95%) ➡️	9.0
[95%, 100%) ➡️	9.5
[100%,…%) ➡️	10.0

Bonus examples

You cannot get a better final grade than 1.0
Your bonus only applies if you pass the final exam with at least 4.0

GOE points ➡️	GOE grade	Exercise score ➡️	Additional GOE points	Final points ➡️	Final grade
49.5 ➡️	4.3	100 % ➡️	10.0	49.5 ➡️	4.3
73.0 ➡️	2.7	93 % ➡️	9.0	82.0 ➡️	2.0
66.0 ➡️	3.0	47.8 % ➡️	4.5	70.5 ➡️	2.7
75.0 ➡️	2.3	49.9 % ➡️	4.5	79.5 ➡️	2.3
79.5 ➡️	2.3	5.0 % ➡️	0.5	80.0 ➡️	2.0
100.0 ➡️	1.0	100 % ➡️	10.0	100.0 ➡️	1.0

Final points = GOE points + Additional GOE points (only if GOE points >= 50)

Exercises as final exam preparation

The exam will be based on the lecture, quiz, homework and group exercises
It focuses on problem solving
It does not include knowledge questions
Statistics of previous years

The statistics show that you can significantly increase your chances to pass the exam by participating in the exercises.

image.png

The following figure shows the average grades (y-axis) for the corresponding exercise performance

(x-axis).