

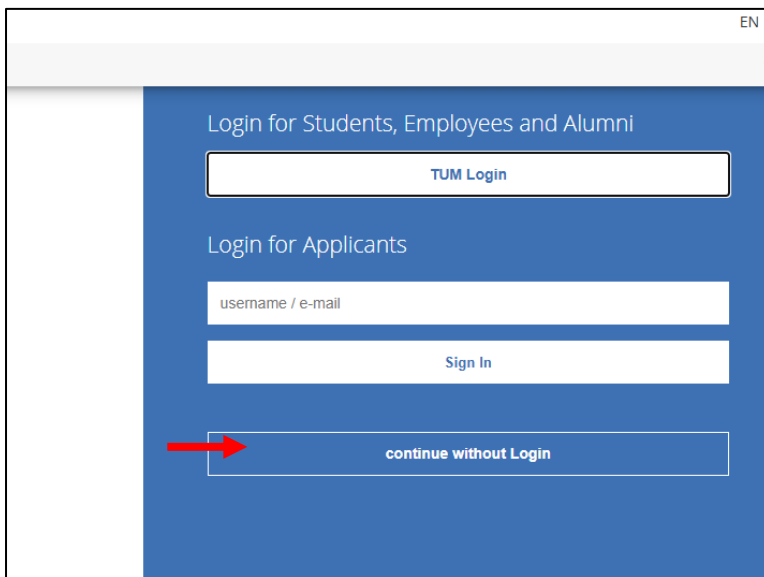
How to Find Courses for Your Exchange Stay at TUM CIT Informatics

When looking for courses, please take our Exchange Program Informatics (98 030) in TUMonline as your starting point (below described as **Option 1**). It contains a vast selection of the most attended courses by exchange students during last years.

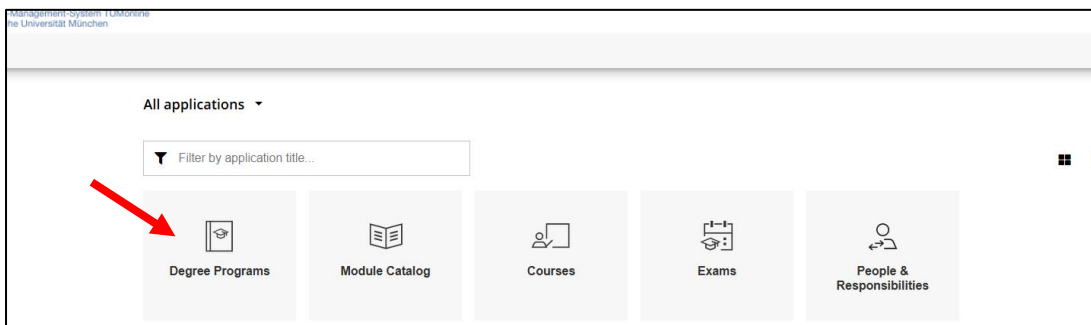
If you would like to look for additional Informatics courses (not listed in 98 030)/courses of other subject-areas, you can use the general course search in TUMonline (below described as **Option 2**).

Option 1 (Exchange Program Informatics):

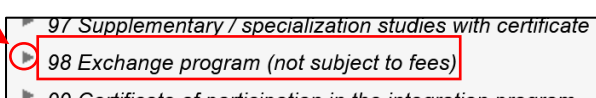
- Go to www.campus.tum.de (TUMonline)
- In the Log In section, click “Continue without login”



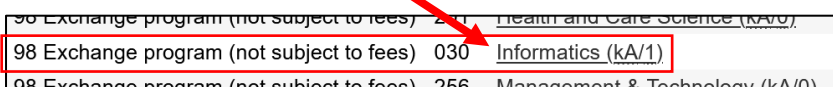
- Select “Degree Programs”.



- Select section “98 Exchange program (not subject to fees)” by clicking on the small grey arrow in front



- You will get a very long list of all the Exchange programs at TUM across many subject-areas.
- Select Exchange program “030 Informatics (kA/1)”.







- You will get a list of the most attended courses by exchange students during last years sorted by type of course (lectures, practical courses, seminars) from which you can choose.

Node filter-Name

- [1] Informatics
 - Lectures in Informatics
 - Algorithms
 - [IN2239] Algorithmic Game Theory
 - Algorithmic Game Theory
 - Algorithmic Game Theory
 - Discrete Structures, Exercise Session (IN0015)
 - [IN2211] Auction Theory and Market Design
 - [IN3410] Selected Topics in Algorithms
 - [IN2007] Complexity Theory
 - [IN2229] Computational Social Choice
 - [IN2003] Efficient Algorithms and Data Structures
 - [IN2004] Efficient Algorithms and Data Structures II
 - [IN2360] Advanced Algorithms
 - [IN2304] Online and Approximation Algorithms
 - [IN0024] Operations Research
 - Computer Graphics and Vision
 - Databases and Information Systems
 - Digital Biology and Digital Medicine
 - Engineering Software-intensive Systems
 - Formal Methods and their Applications
 - Machine Learning and Analytics
 - Computer Architecture, Computer Networks and Distributed Systems
 - Robotics
 - Security and Privacy
 - Scientific Computing and High Performance Computing
 - Games Engineering
 - Programming
 - Basic Modules in Informatics
 - Basic Modules in Information Systems
 - Final Thesis
 - Practical Courses in Informatics
 - Seminar Courses in Informatics
 - Language Courses

orange plus-sign = module/course title.
green sphere = exam for the course.
red triangle = course itself.

- The green sphere represents the exam. 
- The red triangle represents the course itself. 
- To find out, when the course is offered, choose the **academic year**  2022/23  you are interested in (choose the current academic year or a previous cycle, considering the late publication dates of courses).
- The first column indicates in which term the course is offered: **xxW** = winter semester, **xxS** = summer semester.
- When clicking on the underlined course title, you get redirected to the course description.
- Click on the given location for a redirect to a map of the building. TUM has several campuses spread over Munich ([Campus locations](#)), TUM CIT Informatics is mainly located in Research Center Garching:

[IN2346] Introduction to Deep Learning		
[IN2346] Introduction to Deep Learning		
Introduction to Deep Learning (IN2346)		

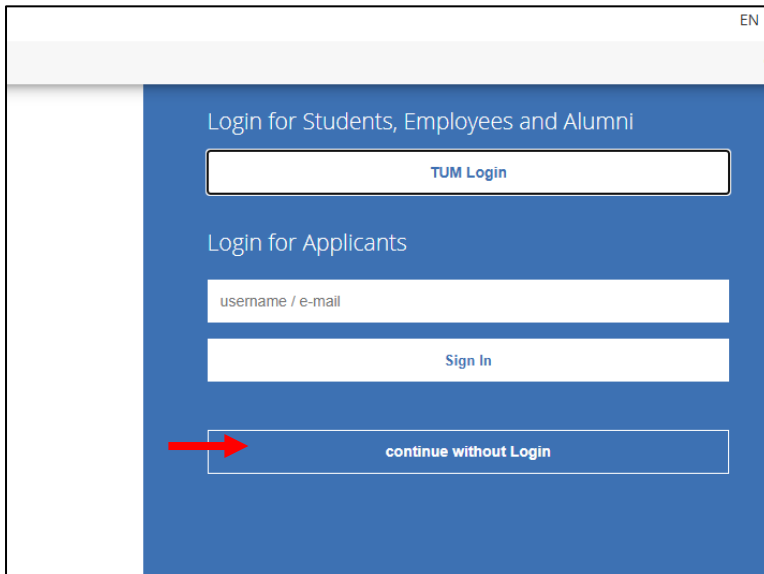
Course(s) in academic year	Part	Lecturer (Assistant)	Place (1st session)	Time (1st session)
0000000673 22W 4SWS VI Introduction to Deep Learning (IN2346)	⊗	Dai A [L], Chen Y, Dahnert M, Dai A, Huang J	00.02.001, MI HS 1, Friedrich L. Bauer Hörsaal (5602.EG.001)	18.10.22 14:00 - 16:00
0000002767 23S 4SWS VI Introduction to Deep Learning (IN2346)	⊖	Nießner M [L], Chen Y, Dahnert M, Franzmann A, Gafni G, ...	00.02.001, MI HS 1, Friedrich L. Bauer Hörsaal (5602.EG.001)	17.04.23 14:00 - 16:00

- Some lectures have accompanying exercises, which are listed under a separate red triangle.

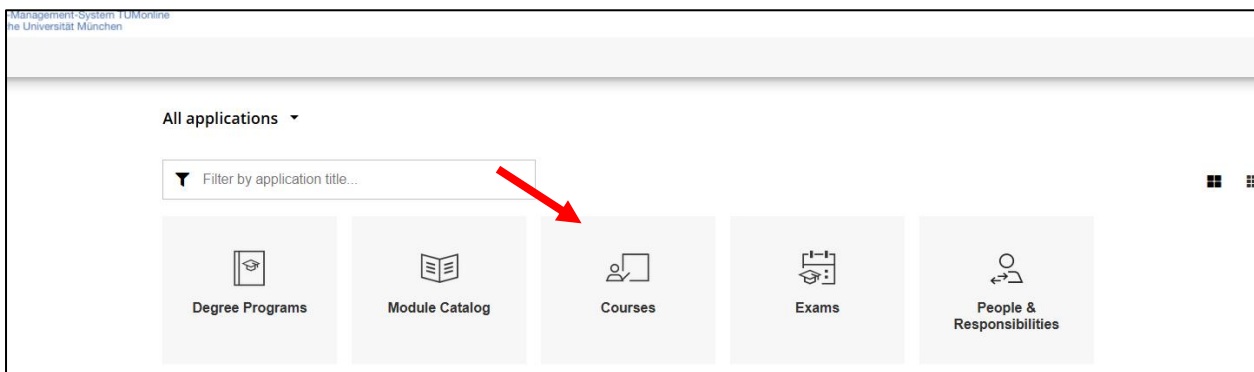
Option 2 (TUMonline's General Course Search):

If you want to choose courses that were not listed within Option 1, you can also check the general course search:

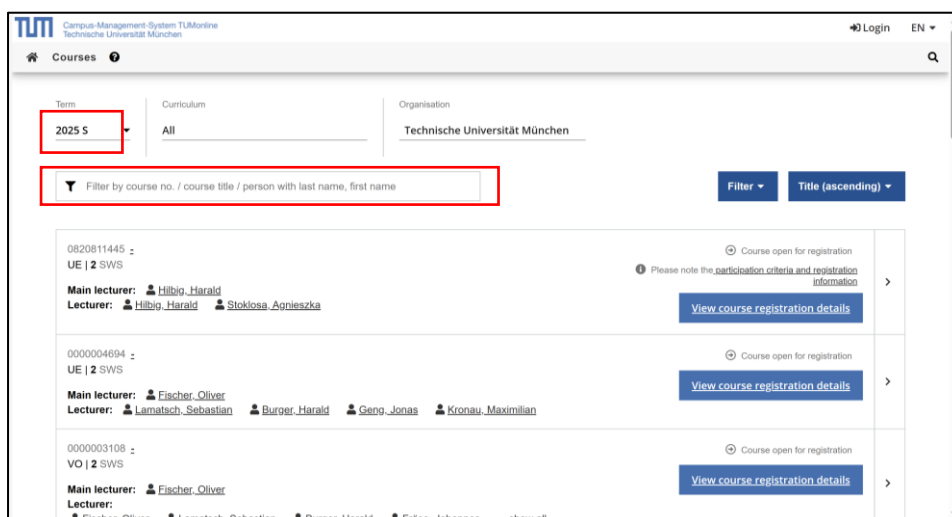
- Go to www.campus.tum.de (TUMonline)
- In the Log In section, click **"Continue without login"**



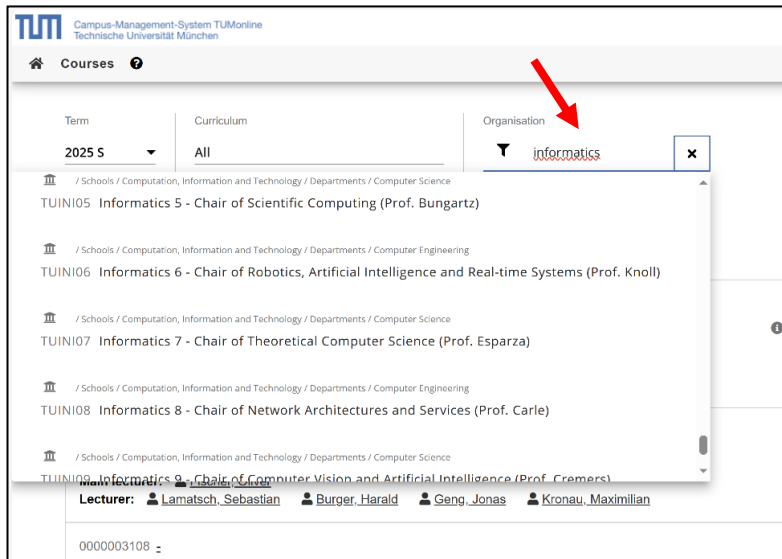
- Select **„Courses“**



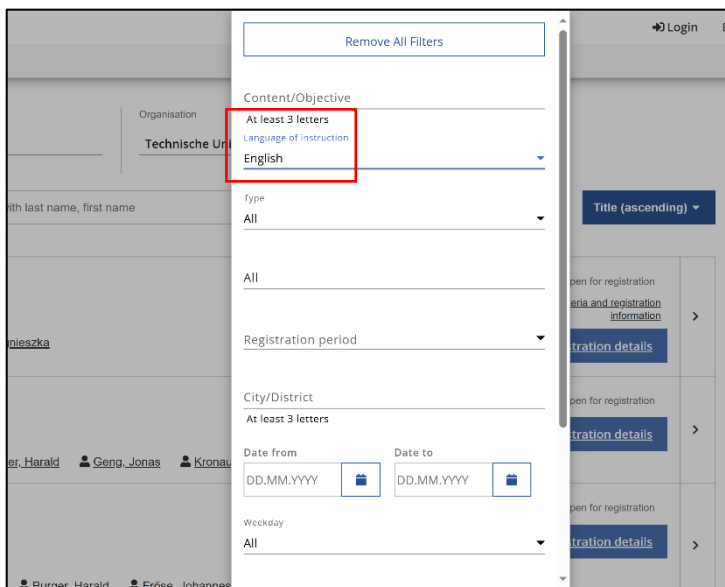
- If the course offer for your exchange semester is not released yet, you have to select the course offer of the previous winter or summer term. We cannot guarantee that all courses will be offered in the same manner, but the previous academic year is always a good indicator for the upcoming course offer.
- You can also use the search bar for entering key words/course topics that might interest you.



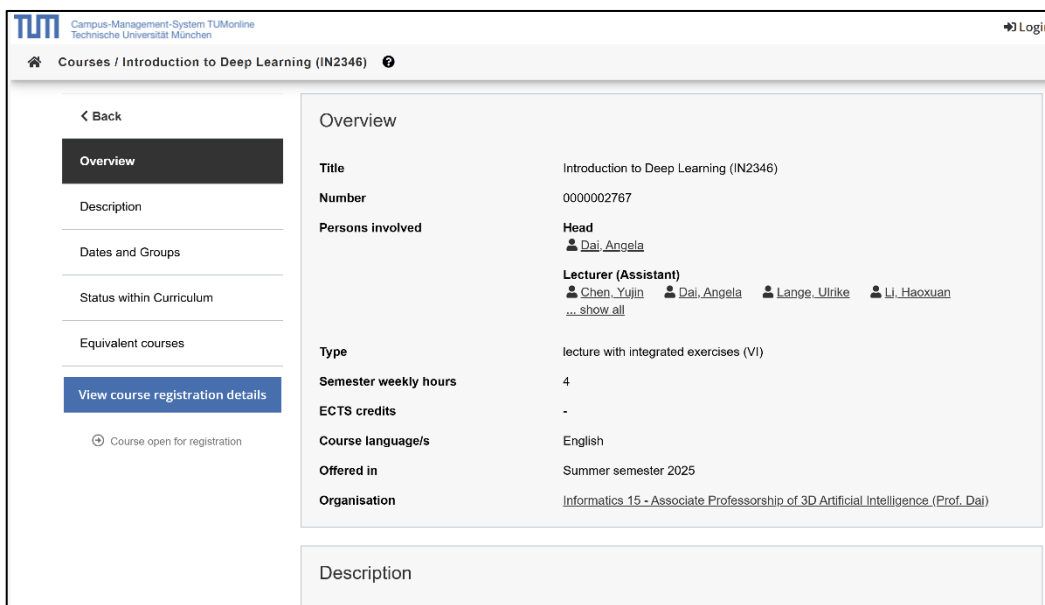
- If you are interested in the course offer of a specific subject-area/chair you can filter it via **“Organization”**.



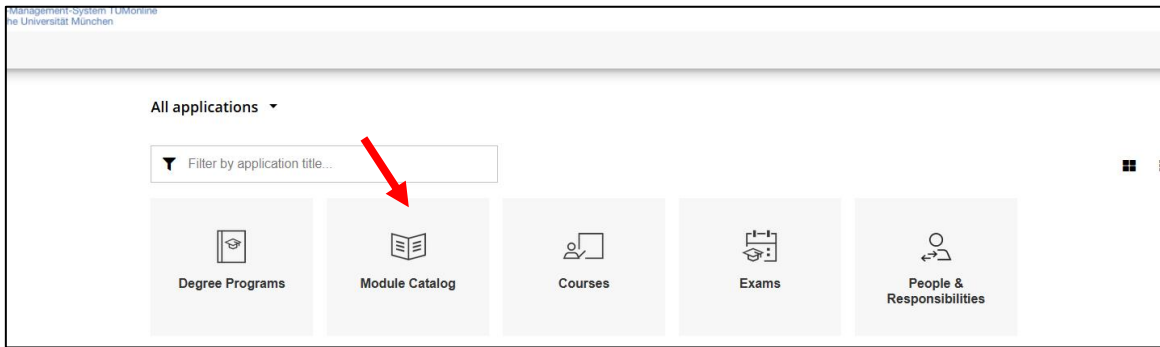
- To reduce the search results to courses that are only taught in English, you can click on **“Filter”** to set the **“Language of Instruction”** to English.



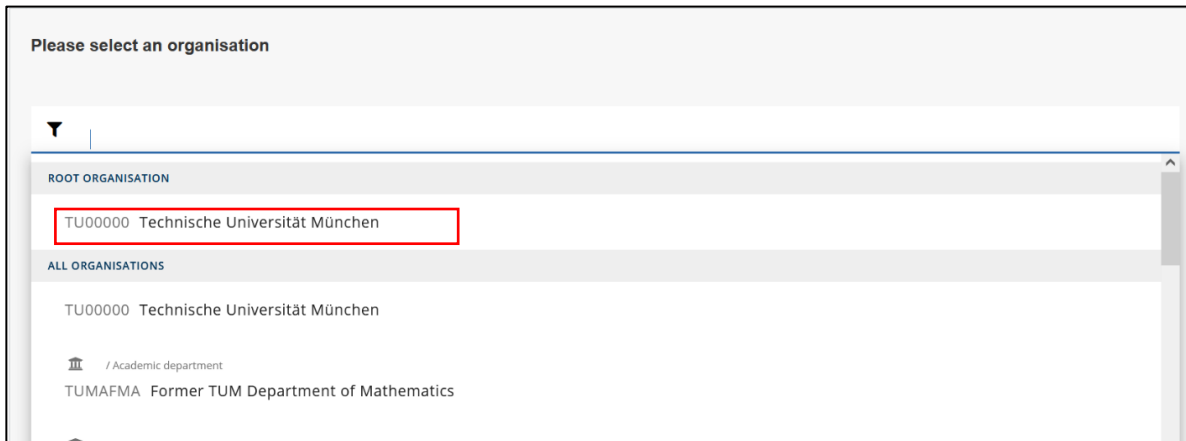
- When you click on a course, the respective course description will open.



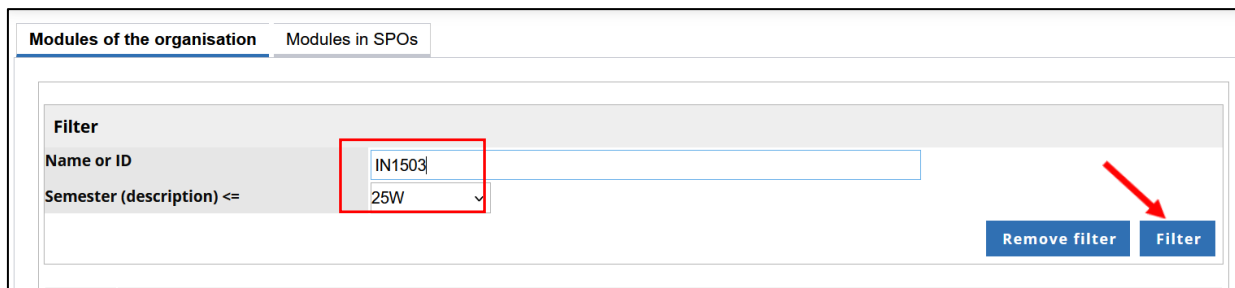
- In case there is not sufficient information in the course description (e.g. no number of ECTS indicated), you can search for more information on single courses selecting **“module catalog”**:



- Choose **“Technische Universität München”** as **“Root Organization”**.



- Set the filter to the semester you’re searching for (or a previous one) and enter the course code/title of the course you would like to search for.



- After clicking on the course title, the module description opens. It provides you with a lot of additional information on the respective course, e.g. ECTS, examination, prerequisites, learning outcomes etc.

Module details	
Name	Advanced Programming
Organisation	Department of Computer Science
Organisation ID	TU51DP2
Comment	
Credits	5
Weighting factor	1
Duration [Acc. to SPO version]	4
Module ID	IN1503
Study and examination performance	
Description of Achievement and Assessment Methods	<p>The assessment consists of a final exam (75min) and of an optional bonus assignments (project work).</p> <p>In the final written exam, through free text, code analysis, and short code development questions, participants can demonstrate that they are able to develop scientific computing software in C++, balancing efficiency and maintainability. They demonstrate that they can choose fitting built-in types and decompose problems into functions, they can explain the basic memory layout and apply appropriate C++ techniques for dynamic resource management, they can solve common programming problems using object-oriented techniques, they can apply generic programming techniques to reduce code duplication, they can apply common performance evaluation modeling, and optimization techniques and can analyze and compare different implementations with respect to their performance. They are additionally given excerpts from the online C++ documentation (which they use throughout the semester) and they use these excerpts to demonstrate that they can make use of fitting STL algorithms in their programs. Finally, via multiple-choice questions, they demonstrate that they can recall and compare development tools essential for working with complex scientific software projects.</p> <p>A hand-written sheet of A4 paper is allowed as a course summary in the final exam.</p> <p>In the bonus project assignment, the participants demonstrate that they can apply the concepts of the course to develop small programming projects collaboratively, in pairs, or by themselves, using development tools essential for working with complex scientific software projects.</p> <p>The final grade is defined by the final exam grade and by the optional bonus assignments (pass/fail, 0.3 grade benefit, provided a passing exam grade).</p> <p>Competence in any programming language. Familiarity with statically-typed languages (e.g. C++, C, Fortran, Java) and/or object-oriented languages (e.g. C++, Java, Python) is helpful.</p>
Prerequisites (recommended)	
Intended Learning Outcomes	<p>After successful completion of this module, participants are able to develop software for scientific computing in C++, balancing performance and maintainability.</p> <p>More specifically, they are able to:</p> <ul style="list-style-type: none"> - choose fitting built-in types and decompose problems into functions; - list prominent software failures (frequently with crucial consequences) in Computational Science related to data types and their usage and are able to explain the sequence of events as well as their causalities; - explain the basic memory layout and apply appropriate C++ techniques for dynamic resource management; - solve common programming problems using object-oriented techniques; - apply generic programming techniques to reduce code duplication; - apply common performance evaluation, modeling, and optimization techniques and can analyze and compare different implementations in respect to their performance; - make use of given parts of the C++ documentation to choose and apply fitting STL algorithms in their programs; and - use development tools essential for working with complex scientific software projects.