Tallinn University of Technology
School of Information Technologies

IAX0583 Programming 1

Extended syllabus
Autumn 2026

Course aims/objectives:	- to develop both creative and analytical thinking through solving original tasks; - to introduce algorithms, means to present them and strategies of creating them, numeral systems, ways to store and process various data types in computers; - to learn to implement algorithms in high-level programming language such as C, Python etc.; - to introduce various software development environments.
Learning outcomes:	Having finished the study of the subject the student: - recognizes various numeral systems and is able to convert numbers from one system to another; - describes the data that is necessary for solving a task, such as: 1. parameters, 2. results, 3. variables - constructs an algorithm that solves a task; - writes a piece of software in C that implements said algorithm; - separates a task into subtasks; - uses subprograms (functions) in software development.
Brief description of the course (topics):	Definition of an algorithm, ways to present and construct algorithms. Separating a task into subtasks. Languages. Language classification. Algorithmic languages. Computer architecture (software layer). Program life-cycle. Command line interface, command line parameters. Operator precedence. Conditionals and control flow statements (including several conditions). Loops with condition checking before and after the statement. Positional and non-positional numeral systems. Converting numbers from one numeral system to another. Fixed point and floating point numbers. Pseudorandom numbers. Declaring variables. Global and local variables. Various data types and their representation in a computer. Converting data from one type to another. Declaring and indexing of arrays. Multidimensional arrays. Operations with arrays (searching, sorting, statistical analysis). Strings and string operations. Formatting results. Time and date based tasks. Creating subroutines, exchanging data between them (parameters, return values). Calling subroutines. Variable scope and lifespan. Files, accessing files for reading and writing.
Language of the course:	Estonian, English
ECTS credits:	6 ECTS
Students:	This is a compulsory course for students studying on IACB, EAAB and MVEB programmes.
Special needs:	Persons with disabilities can participate in this course. Please inform the professor(s) in the beginning of the course of any special instruction, or assessments of this course that may be necessary to enable you to fully participate in this course.
Registration:	Students who would like to take the course should declare the course in the ÕIS (Student Information System) by deadlines set in the academic calendar.

Prerequisite courses and/or knowledge:	Basic computer literacy
Prerequisite resources:	Access to a computer and internet for completing individual work. Software to write documentation, e.g. OverLeaf, Libreoffice, etc. Modelling tools that support UML, e.g. Visual Paradigm Community Edition. Code- or text editor, e.g. Geany, Vim, etc. GNU/Linux, distribution preference is left to the student. Tutorials and guidance is provided for debian-based distributions. Various software packages and tools for programming and supporting development.
Professor(s):	Risto Heinsar, risto.heinsar@taltech.ee
Contacting Professor(s):	Preferred means of contact is by sending a direct message Mattermost on the university's private server https://mm.ttu.ee, response is provided within 3 wokdays, typically within 1 calendar day. If there is no response within three working days, reply to your previous message to highlight the conversation. „No-Hello" principle applies, i.e. write about the reason for the message in full alongside hello, so that I'd have all the required information to reply to you with a complete response.
Schedule for classes:	According to the official university timetable on tunniplaan.taltech.ee. Classes may be cancelled or moved to alternative time due to national holidays or illnesses. Students will be notified when this happens on Mattermost. Work is planned on-site. Lectures are recorded, recordings can be accessed on Moodle, assuming the classroom supports Echo360.
Study process description:	There are a total of 14 lessons planned that use a combined lecture + practice format. Each lesson starts by introducing the practical knowledge required to solve the tasks. The time following the lecture is planned for solving and defending of the lab tasks. Most topics are built on the topics preceeding it, i.e. it's important to attain the knowledge from the previous lessons. Practical task deadlines use the sliding window principle. Each student is given two homework tasks, which need to be submitted alongside a report by the deadline specified. Homeworks both reinforce and expand on the knowledge from the classes.
Course's e-support:	Course materials can be accessed via the e-learning environment Moodle under the course title IAX0583 Programming 1 https://moodle.taltech.ee/course/view.php?id=37670. Students can enroll to the course themselves, password is listed on the enrollment page. All materials are accessible without authentication and limitations at https://blue.pri.ee/ttu/
Study literature:	Main location of study materials: https://blue.pri.ee
Supplementary materials:	The C Programming Language, 2nd Edition 2nd Edition. Brian W. Kernighan, Dennis M. Ritchie, Pearson, 1988. Beej's guide to C programming: https://beej.us/guide/bgc/
Assessment method:	Exam

ASSESSMENT METHOD	ASSESSMENT CRITERIA
Written homework	*Each student is given three individual assignments with a combined total of 200 points (100 + 100 points).* *The points are a part of the final grade.*
Test	*There is one test during the semester, graded on a scale of 150p.* *The points are a part of the final grade.* *The points are a part of the exam prerequisite.*
Lab tasks	*The total number of points from lab tasks is 250.* *The points are a part of the final grade.* *The points are a part of the exam prerequisite.*
Written examination	*Written examination is graded on the scale of 400 points.* *The total points for the course consist of the points for the written examination, homework (max 200p), test (max 150p) and labs (max 250p).*
PREREQUISITES FOR WRITTEN EXAMINATION	*Student has received a positive result from the test.* *Student has accumulated at least 50% of the points for lab tasks and the test.*

Final grade:

The sum of points for each item is converted into a grade using the following principles:
"5" excellent 901-1000
"4" very good 801-900
"3" good 701-800
"2" satisfactory 601-700
"1" poor 501-600
"0" fail less than 501

Academic integrity:

As a student at Tallinn University of Technology, you have an obligation to conduct your academic work with honesty and integrity according to University standards. It is expected that all work that you submit will be your own, and that you have actually done the work that you are submitting. Plagiarism and cheating will not be tolerated. Should you be found to be guilty of such activities, it will be followed with grade "0" for the assignment/exam and a notice will be filed to the School's Committee for Handling Violations of Academic Practice and Contemptible Behaviour. Depending on the Committee's proposal, it may lead to Dean issuing a letter of reprimand or in case of repeated or very severe misconduct, exmatriculation from the University.

Use of artificial intelligence

You are allowed to use generative AI for rephrasing, translating and fixing your own written work (E.g. Homework report). Large-scale use of generative AI output is not permitted. For referencing, see https://haldus.taltech.ee/sites/default/files/2024-04/Guide%20to%20using%20sources%2016042024.pdf

In other aspects of the subject, use of AI is permitted for
- composing short snippets of code
- using it as a mentor / assistant, asking it for explanations on existing work or work you've written, finding bugs, gathering ideas

Use of AI is phrohibitted:

- During exams, tests, and quizzes
- When defending tasks
- For writing complete solutions to tasks

The student is responsible for all AI created and altered content. Students must check all the created content and be able to defend it.

More detailed overview of rules regarding use of AI are written in the study guide.

Detailed schedule and topics

The semester plan is preliminary and might change in case of cancellations, changes in available material, etc. Most up to date version is availble on https://blue.pri.ee/ttu/.

Week 1
Introduction
Overview of the study arrangement, introduction to subject, introduction to the tools used, properties of programming languages, C programming language, code statement, code block, coding style, library, input-output statements, variables, data types, commenting code, conditionals, properties of algorithms, UML, introduction to modelling.

Week 2
Conditionals
Conditional statements, logical operators, truth table, De Morgan's law, floating point, nesting conditionals.

Week 3
Loops
Preprocessor, macros, magical numbers, compiling from source to machine code, loops and their properties, including exit- and entry control.

Week 4
Functions
Formatting numbers in output, function definition, structure, return value, side effect, prototype, argument, parameter, call, variable scope and lifetime, global and local variables.

Week 5
Arrays
Type casting, floating point precision, math library, one dimensional arrays, initialization, sizeof operator, linters.

Week 6
Arrays
Using the debugger.

Week 7
Sorting
Different number systems, including binary and hexidecimal, positional and non-positional number systems, converting integers between number systems, integer overflow, bit, byte, nibble, signed and unsigned integer, MSB, LSB, sign bit, simple sorting algorithms, including bubble sort.

Week 8
Standard streams, pseudo-random numbers
Process standard streams, redirecting streams, text file, directory traversal, listing and executing programs in CLI, booleans, single line code blocks, dangling else, randomness and pseudo-random numbers, UNIX time.

Week 9
Matrix
Two-dimensional arrays, encoding values, 64-bit integers.

Week 10
Menu-programs
Character, string, menu program.

Week 11
Test
During the lesson, a test is written for the material covered up to this point.

Week 12
Linux and CLI
UNIX directory structure, importance of CLI, common commands, file permissions and properties, viewing, filtering and editing text files on CLI, manually compiling from CLI, piping, SSH

Week 13
Strings
String, character, encoding characters, ASCII, Unicode, input buffer, buffer overflow attacks, processing characters and strings, CSV.

Week 14
Command line arguments
Using command line arguments, converting strings.

Week 15
Files
Using external resources, text and binary files, reading and writing text files, input and output buffers, file pointer, relative and full path.

Week 16
Spare time
Spare time is commonly used for the first exam time, rewriting the test and performing defences for the labs and homework.