

C Programming Language

Tomáš Krajník

Department of Computer Science,
Faculty of Electrical Engineering,
Czech Technical University

Overview of the Lecture

- **Part 1 – Course Organization**
 - Course Goals
 - Means of Achieving the Course Goals
 - Evaluation and Exam
- **Part 2 – Introduction to C Programming**
 - Program in C
 - Values and Variables
 - Expressions
 - Standard Input/Output

Part 1: Course Organization

- Course Goals
- Means of Achieving the Course Goals
- Evaluation and Exam

BE5B99CPL – C Programming Language

- Course web page
 - <https://cw.fel.cvut.cz/wiki/courses/be5b99cpl>
- Homework submission
 - Individually during the seminars
- Lecturer
 - Tomáš Krajník
 - <https://scholar.google.com>
 - Department of Computer Science
 - <http://cs.fel.cvut.cz>
 - Artificial Intelligence Center (AIC)
 - <http://aic.fel.cvut.cz>
 - Chronorobotics laboratory
 - <http://chronorobotics.fel.cvut.cz>

Lecturers

Tomáš Krajník

Jiří Ulrich

Jiří Janota

<https://scholar.google.com>



- **Chronorobotics**

- understanding of time by autonomous systems
- Coordinator or principal investigator of several EU research projects
- e.g., sensorbees.eu, roboroyale.eu

- Department of Computer Science

- Artificial Intelligence Center (AIC)

- Chronorobotics laboratory

- <http://cs.fel.cvut.cz>

- <http://aic.fel.cvut.cz>

- <http://chronorobotics.fel.cvut.cz>

Some highlights

- **2024:** our research on the cover of the Science Robotics
- **2024:** Jiří Janota: 2nd best IT-related master thesis in Czechia and Slovakia (IT SPY Award)
- **2022:** Jiří Ulrich: The best IT-related master thesis in Czechia and Slovakia (IT SPY Award)



Course Goals

- Master (yourself) programming skills
 - Seminars, homeworks, microtests, mid-term test, test
- **Acquire knowledge** of C programming language
- **Acquire experience** of C programming to use it efficiently
- Gain experience to read, write, and understand small C programs
- Acquire programming habits to write
 - easy to read and understandable source codes and reusable programs.
- Experience programming with desktop computers
- Usage services of the OS: input/output, file read/write.
- Learn to write multithreading applications

Course Organization and Evaluation

- BE5B99CPL– C Programing Language
 - Extent of teaching: 2(lec) + 2(lab) + 5(hw)
 - Completion: Z,ZK; Credits: 6
 - Z – ungraded assessment, ZK – exam
-
- work during the semester – homeworks, microtests and mid-term test (Z)
 - test and implementation exam (ZK)
-
- attendance to labs, submission of homeworks, microtest results

Outline

- Course Goals
- **Means of Achieving the Course Goals**
- Evaluation and Exam

Resources and Literature

- Main course textbook:
 - C Programming: A Modern Approach, 2nd Edition, K. N. King, W. W. Norton & Company, 2008, ISBN 860-1406428577
- Lectures – support for the textbook, slides, comments, and your notes
 - source codes are provided as a part of the lecture materials, hands-on coding during lectures
- Laboratory Exercises
 - gain practical skills by doing homeworks, microtests, comments, and your notes

Further Reading

- Programming in C, 4th Edition
 - Stephen G. Kochan, Addison-Wesley 2014,
 - ISBN 978-0321776419
- 21st Century C: C Tips from the New School
 - Ben Klemens, O'Reilly Media, 2012,
 - ISBN 978-1449327149
- The C Programming Language, 2nd Edition, ANSI C
 - Brian W. Kernighan, Dennis M. Ritchie, Prentice Hall, 1988
- Advanced Programming in the UNIX Environment, 3rd edition
 - W. Richard Stevens, Stephen A. Rago, Addison-Wesley, 2013
 - ISBN 978-0-321-63773-4

Further Reading

- The C++ Programming Language, 4th Edition(C++11)
 - Bjarne Stroustrup, Addison-Wesley, 2013
 - ISBN 978-0321563842
- Introduction to Algorithms, 3rd Edition, Cormen
 - Leiserson, Rivest, and Stein, The MIT Press,,2009,
 - ISBN 978-0262033848
- Algorithms, 4th Edition
 - Robert Sedgewick, Kevin Wayne, Addison-Wesley, 2011
 - ISBN 978-0321573513

Lectures – Winter Semester Academic Year 2025/2026

- Schedule for the academic year 2025/2026
 - <http://www.fel.cvut.cz/en/education/calendar.html>
- Lectures:
 - Karlovo náměstí, Room No. KN:A-420, Wednesday, 9:15–10:45
 - 14 teaching weeks
 - 12 lectures + 1x mid-term test + (1x final test or 1x additional lecture)

Overview of the Lectures

1. Course information, Introduction to C programming - King: chapters 1, 2, and 3
2. Writing your program in C, control structures (loops), expressions - King: chapters 4, 5, and 6
3. Data types, arrays, pointer, memory storage classes, function call - King: chapters 7, 8, 9, 10, and 11
4. Data types: arrays and strings - King: chapters 11, 12, and 13
5. Data types: arrays and pointers - King: chapters 11, 12, and 13
6. Data types: Struct, Union Preprocessor and Large Programs - King: chapters 14, 15, and 16
7. Input/Output – reading/writing from/to files - King: chapters 21-27
8. Selected topics from previous lectures based on feedback from the students
9. Mid-term test (implementation exercise)
10. Multi-thread application models, POSIX threads and C11 threads
11. ANSI C, C99, C11 – differences and extensions
12. Differences between C and C++: Introduction to object oriented programming in C++
13. Object oriented programming in C++: classes, objects, encapsulation, and polymorphism
14. Homework evaluation, early exam, additional lecture

Lecturers

Jiří Ulrich

Jiří Janota

<https://scholar.google.com>

- Department of Computer Science
- Artificial Intelligence Center (AIC)
- Chronorobotics laboratory



- <http://cs.fel.cvut.cz>
- <http://aic.fel.cvut.cz>
- <http://chronorobotics.fel.cvut.cz>

Communicating Any Issues Related to the Course

- Ask the lab teacher or the lecturer



- Use e-mail for communication: `name.surname@fel.cvut.cz`
 - Use your faculty e-mail
 - Put CPL or BE5B99CPL to the subject of your message
 - Send copy (Cc) to lecturer

Computers and Development Tools

- Network boot with home directories (NFS v4)
- Data transfer and file synchronizations
 - ownCloud, SSH, FTP, USB
- Compilers: **gcc** or **clang**
 - <https://gcc.gnu.org> or <http://clang.llvm.org>
- Project building: **make** (GNU make)
 - Examples of usage on lectures and labs
- Text editor – vim, gedit, atom, sublime + tmux
 - <http://www.root.cz/clanky/textovy-editor-vim-jako-ide>
- “No IDE” approach
 - <http://c.learnthecodehardway.org/book/ex0.html>
- Debugging – **gdb**, **cgdb**, **ddd**

Services – Academic Network, FEE, CTU

- General information
 - <http://www.fel.cvut.cz/cz/user-info/index.html>
- Cloud storage ownCloud
 - <https://owncloud.cesnet.cz>
- Sending large files
 - <https://filesender.cesnet.cz>
- Schedule, deadlines – FEL Courseware
 - <https://https://cw.fel.cvut.cz>
- Gitlab FEL
 - <https://gitlab.fel.cvut.cz/>
- Academic and campus software license
 - <https://download.cvut.cz>

Homeworks

- Four homeworks with a total point value of 30 with 20 as a minimum
 - robotics - themed, but no knowledge or robotics required
 - 10% penalisation for every day of delay
 - submission through brute
1. Iterative calculation with a provided formula
 - probabilistic update via a Bayes filter
 2. Updating a fixed-sized array - calculations over a 2D array of values
 - occupancy grid construction from a laser rangefinder
 3. Operations over variable-sized array - reallocation and reindexing of an array
 - occupancy grid construction from a robot-carried laser rangefinder
 4. Multi-process operations over a shared array
 - occupancy grid construction from a multi-robot team equipped with laser rangefinder

Microtests

- Six microtests with a total point value of 15 with 6 as a minimum
 - aimed at basic concepts
 - 5 - 15 minutes at the start of labs 1, 2, 4, 6, 8 and 10
1. Example test with 0 point gain
 - return the number of command line arguments
 2. Command line argument handling
 3. Advanced calculation
 4. Array or string operations
 5. Input and output, file handling
 6. Input and output, file handling, array operation, search

Mid-term and final exam tests

- Mid-term test with a total point value of 15 with 5 as a minimum
 - will be preceded by a lecture with difficult concepts based on student feedback
 - 180 minute-long test submitted to brute
- Exam theoretical test with a total point value of 15 with 5 as a minimum
 - 60 minute theoretical test with simple questions
 - question pool will be available
- Exam implementation test with a total point value of 25 with 10 as a minimum
 - 240 minute-long test submitted to brute with on-the fly evaluation

Expected results

- | | |
|--|-------------|
| 1. All homeworks work and submitted before the deadlines | - 30 points |
| 2. Microtests passed | - 15 points |
| 3. Mid-term test passed | - 15 points |
| 4. Exam theoretical test passed | - 15 points |
| 5. Exam implementation test passed | - 25 points |

Course Evaluation

Category	Maximum points	Required minimum points
Homeworks	30	20
Microtests	15	6
Mid-term test	15	5
Exam - theory	15	5
Exam - implementation	25	10
Overall (not just a sum)	100	50

To gain an ungraded assessment, all homeworks must be submitted.
Homeworks, microtest and mid-term test must reach or exceed the threshold.

Grading

Grade	Point range	Evaluation
A	90 - 100.0	Excellent
B	80 - 89.99	Very good
C	70 - 79.99	Good
D	60 - 69.99	Satisfactory
E	50 - 59.99	Sufficient
F	0.0 - 49.99	Failed