

The University of Aizu
School of Computer Science and Engineering

Programming in C

Course Syllabus
(Special Track for Foreign Students)

Contents

Programming in C.....	3
Course Description.....	3
Prerequisites and Dependencies.....	3
Objective	3
Learning Outcomes	3
Topics	4
Evaluation.....	4
Referential Sources	4
Textbooks	4
Useful Books and Papers.....	5
Useful Web Sources	5

Programming in C

Course Syllabus (Special Track for Foreign Students)

This course is aimed at advancing concepts of programming and software code organization within the framework of structural and procedural programming paradigms. The special track is organized as a series of lectures, hands-on workshops and exercises using C programming languages and focusing on discussing how to write a program of moderate complexity by using C language.

Course Description

- Target audience:
 - First year students of the Department of Computer Science and Engineering
- Course Period: One quarter
- Total Class Hours: 48 hrs (24 periods)
 - Lectures: 16 hrs
 - Hands-on workshops: 16 hrs (practical sessions supervised by the lecturer and TAs)
 - Exercises: 16 hrs (individual projects assisted by TAs)
- Credits: 4.0
- Lecturer: Evgeny PYSHKIN, Senior Associate Professor, Software Engineering Lab

Prerequisites and Dependencies

This course is based on the course “Introduction to Programming”. So the students must have basic knowledge of mathematical and algorithmic logics, to understand major control structures such as branching, loops and expressions, to be able to use functions and to create arrays of elementary objects in their simple C programs.

The course teaching language is English, so students have to have communication, reading and apprehension skills of English.

Objective

The course is oriented to those who want to advance structured and procedural programming understating and to improve C programming skills. The major objective is to provide students with understanding of code organization and functional hierarchical decomposition with using complex data types.

Learning Outcomes

After course completion the students will have the following learning outcomes:

- Understanding a functional hierarchical code organization.
- Ability to define and manage data structures based on problem subject domain.
- Ability to work with textual information, characters and strings.
- Ability to work with arrays of complex objects.
- Understanding a concept of object thinking within the framework of functional model.
- Understanding a concept of functional hierarchical code organization.
- Understanding a defensive programming concept. Ability to handle possible errors during program execution.

Topics

Introduction to the course. Brief summary of the concepts and models presented in the course “Introduction to Programming”. Programming vs. software development revisited. Major stages of software production. Comment to learning process organization: “orthogonality” of lectures/hands-on/exercise tracks.

Characters and strings. String library. Other elements of standard library. Importance to use standard library functionality.

“Algorithms + Data Structures = Programs” revisited. What if the program objects are more complex against elementary types? From subject domain to data models. Structures. Functional hierarchical code organization with respect to structural types.

Input/output revisited. Working with files.

Arrays of structural objects. Dynamic memory allocation. Advanced introduction to scope and memory classes. Functions dealing with complex types. C program organization revisited: attention to focus on data types.

A concept of defensive programming: “Garbage in” shouldn’t mean “garbage out”. Error handling in C programs.

Linked types by an example of linked lists. Introduction to other containing structures. Pointers advanced. Why pointers might be dangerous. Safe and unsafe code.

Final discussion.

Evaluation

The final grade will be calculated based on the following weights:

- Tests and quizzes during lecture and hands-on time – 25%
- Individual projects – 30%
- Bonus points for active participation in hands-on workshops – 20%
- Final test – 25%

Test and quizzed during lectures are also used as students’ attendance confirmation. Students whose attendance is lower than regulated attendance rate (2/3 or more) are considered to abandon the class. Students who didn’t achieved at least 50% progress before final test are also considered to abandon the class.

In contrast, students who successfully performed their individual projects with a progress level higher than 80%, and demonstrated good results during regular classes may be allowed by the lecturer not to take the final test with automatically achieving the maximum score for the final test.

Referential Sources

The list of referential sources is subject of further updates.

Textbooks

To be defined

Useful Books and Papers

1. Greg Perry, Dean Miller. "C Programming Absolute Beginner's Guide", 3rd ed., Que Publishing, 2013.
2. Stephen G. Kochan. "Programming in C", 4th ed., Addison-Wesley Professional, 2014.
3. Paul Deitel and Harvey Deitel. "C for Programmers with an Introduction to C11", Prentice Hall, 2013.
4. Stephen Prata. "C Primer Plus", 6th ed., Addison-Wesley Professional, 2013.
5. Samuel P. Harbison and Guy L. Steele Jr. "C: A Reference Manual", 5th ed., Pearson, 2002.
6. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein. "Introduction to Algorithms", 3rd ed. MIT Press, 2009. Info here: <https://mitpress.mit.edu/books/introduction-algorithms>
7. Jeffrey McConnell. "Analysis of Algorithms", 2nd ed., Jones & Bartlett Learning, 2007.
8. Steve McConnell. "Code Complete", 2nd ed. Microsoft Press, 2004.
9. Charles Petzold. "Code: The Hidden Language of Computer Hardware and Software", Microsoft Press, 2000.
10. Jon L. Bentley. "Programming Pearls", 2nd ed. Addison-Wesley, 1999.
11. Brooks F. The Mythical Man-Month. Essays on Software Engineering. Anniversary Edition. Addison-Wesley, 1995.

Useful Web Sources

1. Course page (to be organized)
2. <http://web-int.u-aizu.ac.jp/~pyshe/> : Evgeny Pyshkin's web page on the university web site.
3. <https://www.programiz.com/c-programming> : Learn C Programming. The definitive guide
4. <http://www.cprogramming.com/> : C Programming and C++ Programming
5. <http://web-ext.u-aizu.ac.jp/course/prog1/> (in Japanese) : University of Aizu "Programming C" course home page.
6. http://kspt.icc.spbstu.ru/media/files/people/pyshkin/books/AlgDataStrA5_2009.pdf : "Data Structures and Algorithms: Implementation in C/C++" by Evgeny Pyshkin (in Russian, but the source code examples may be useful)