The University of Aizu
School of Computer Science and Engineering

Introduction to Programming

Course Syllabus
(Special Track for Foreign Students)

Evgeny Pyshkin, Senior Associate Professor

27.01.2017

Contents

Introduction to Programming

Course Syllabus (Special Track for Foreign Students)

This course is aimed at presenting foundation concepts of programming and software code organization. The special track is organized as a series of lectures, hands-on workshops and exercises using C programming languages and focusing on discussing structural programming concepts.

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

Since this is a 1st year course, there is no particular entry requirements. However, students are expected to have elementary knowledge of mathematical and algorithmic logics and to be able to learn how to understand programming code. Some primary understanding of computer organization basics is desirable.

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

In turn, this course provides a foundation for the course "C Programming" which is the second part of the 1st programming track.

Objective

The course is oriented to those who want to learn programming basics with using C programming language as an implementation tool. The major objective is to provide students with understanding of programming essentials, including algorithms, data types, elementary control structures and functions used within the framework of imperative and structural programming paradigms.

Learning Outcomes

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

- Understanding foundation concepts of information and information processing in computer systems: a matter of information, data representation, coding systems.
- Understanding of an algorithm and its definition.
- Understanding of a programming language syntax and its definition by example of C language.
- Knowledge of basic principles of imperative and structural programming.
- Ability to write simple programs in C language by using basic control structures (conditional statements, loops, switches, branching, etc.).

- Ability to create a programmable model for a problem given.
- Understanding a function concept and how to deal with function arguments and parameters.
- Ability to use pointers and pointer arithmetic in the simple cases. Basic knowledge of working with arrays in C language.
- Understanding a defensive programming concept. Ability to handle possible errors during program execution.
- Elementary knowledge of programming code style.

Topics

Introduction to the course. Programming, programming models, programming languages. Programming vs. software development. Languages and platforms. Software development process. Why software is complex? Expecting changes. Comment to learning process organization: “orthogonality” of lectures/hands-on/exercise tracks.

Programming fundamentals: information, coding, matter. Numerical systems, significance of binary code. Operations in binary code. Elementary introduction to syntax and semantics. Code translation basics. How to define a language syntax? Naming and formatting: basic knowledge.

“Algorithms + Data Structures = Programs” (Wirth). How to define program objects? How to define actions? Operations and expressions. Understanding a data type concept. Embedded types. What is an algorithm? Ways to define an algorithm.

How C program is organized. Basic structural programming language instructions. Conditional statements. Switching. Iterative code. Possible implementations. From the source code to the executable code: compiling basics.

Input/output library basics. Console applications.

Goto/break/continue statements. “Goto statement considered harmful” (Dijkstra).

Floating point computations. Data types revisited.

Functions. Function arguments and function parameters. Functional decomposition: basic knowledge. Code organization revisited. Compiling revisited.

Pointers and arrays: introduction. General model and particularities of C implementation.

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 that 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. Charles Petzold. “Code: The Hidden Language of Computer Hardware and Software”, Microsoft Press, 2000.

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.