Programming in C
Course Syllabus
(Special Track for Foreign Students)

Evgeny Pyshkin, Senior Associate Professor
27.01.2017
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

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 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


Useful Web Sources

1. Course page (to be organized)