

„POLITEHNICA” UNIVERSITY OF TIMIȘOARA

SYLLABUS for the discipline:

PROGRAMMING TECHNIQUES *Advanced C programming. Problem solving.*

FACULTY OF AUTOMATIC CONTROL AND COMPUTING

DOMAIN / SPECIALIZATION: COMPUTERS AND INFORMATION TECHNOLOGY

Year of studies: I Semester: 2

Course and applications instructor: As.univ.dr.ing. Alin ANTON

Number of hours/week/Evaluation/Credits					
Course	Seminar	Laboratory	Project	Evaluation	Credits
2	0	2	0	Exam	4

A. COURSE OBJECTIVES

Acquiring advanced programming skills with the C programming language. Understanding and being able to apply basic principles of program design and architecture. Understanding and being able to apply basic algorithms for solving problems: Greedy, Backtracking, Dynamic programming, Divide and Conquer. Understanding and being able to apply dynamic memory allocation for handling dynamic lists. Acquiring basic notions about abstract data types.

B. PRECONDITIONS: Computer programming (sem I)

C. COMPETENCES

Cod	Conținut competență	Procent	Credit	
Competențe profesionale				
C1	Operarea cu fundamente științifice, ingineresti și ale informaticii	25	1	
C2	Proiectarea componentelor hardware, software și de comunicații	30	1,2	
C3	Soluționarea problemelor folosind instrumentele științei și ingineriei calculatoarelor	20	0,8	
C4	Îmbunătățirea performanțelor sistemelor hardware, software și de comunicații	5	0,2	
C5	Proiectarea, gestionarea ciclului de viață, integrarea și integritatea sistemelor hardware, software și de comunicații	0	0	
C6	Proiectarea sistemelor inteligente	0	0	
Competențe transversale				
CT1	Comportarea onorabilă, responsabilă, etică, în spiritul legii pentru a asigura rezolvarea problemei	1	0,4	
CT2	Identificarea, descrierea și derularea proceselor din managementul proiectelor, cu preluarea diferitelor roluri în echipă și descrierea clară și concisă, verbal și în scris, în limba română și într-o limbă de circulație internațională, a rezultatelor din domeniul de activitate	0	0	
CT3	Demonstrarea spiritului de inițiativă și acțiune pentru actualizarea cunoștințelor profesionale, economice și de cultură organizațională	1	0,4	
		Total	100	4

D. LECTURES OUTLINE

a) course (28 hours)

- Free software and intellectual property:** Free software philosophy. Intellectual property, the copyright law. Digital piracy. Ethics of computing.
- Advanced concepts in the C programming language:** Structured data types. Bit fields. Functions with variable number of arguments. Command line arguments. Function pointers. The C preprocessor.
- Files:** Generalities. Text files. Binary files. Error handling while working with files.
- Systematic design and development of large programs:** Programming style. The Stepwise Refinement Method.
- Independent compilation:** Preprocessing, compiling and linking. Header files. External and static symbols.

6. **Recursion:** Generalities. The activation record. Recursive functions in C.
7. **General paradigms for designing programs:** Greedy. Backtracking. Divide and Conquer. Dynamic programming.
8. **Dynamic data structures:** Singly linked lists. Ordered lists. Doubly linked lists.
9. **Abstract data types:** Implementing abstract data types in C. The Stack.
10. **Sorting techniques:** Insertion based sorting. Mergesort. Timsort.

b) application subjects (laboratory) (28 hours)

1. Free software, GNU/Linux and the Linux environment.
2. Structures, unions, bit fields. Command line arguments.
3. Function pointers.
4. Processing text files.
5. Processing binary files.
6. The stepwise refinement method. Large programs. The preprocessor.
7. Programming style elements.
8. Recursion.
9. Greedy. Divide and Conquer.
10. Backtracking.
11. Dynamic programming.
12. Linked lists.
13. Abstract data types.
14. Sorting.

E. EVALUATION PROCEDURE

Computer based evaluation	50 %
Lab activity	40 % (40% theory, 50% individual programming applications, 10% homework)
Course activity	10 %

F. METHODOLOGY

Lectures are given with video-projected slides. All course sessions are interactive and challenging and course activity is 10% of the final grade. Laboratory material is derived from the lecture discussions and consists of theoretical tests and individual programming activity. Homework is assigned during each lab and accounts for 10% of the application grade.

G. REFERENCES

1. Ciocârlie Horia, Programming techniques (University library)
2. Samuel P. Harbison, Guy L. Steele Jr., C: A reference manual , <http://careferencemanual.com> (University library)
3. Kenneth A. Reek, Pointers on C
4. Alana, R. Feuer, The C Puzzle Book
5. Steven S. Skiena, The Algorithm Design Manual
6. Steven S. Skiena, Miguel A. Revilla, Programming Challenges
7. Steven Halim, Competitive Programming
8. Miguel A. Revilla, William B. Poucher, From Baylor to Baylor
9. van Lindberg, Intellectual Property and Open Source: A Practical Guide to Protecting Code
10. Steve Oualine, C Elements of Style

H. INTERNATIONAL COMPATIBILITY

Purdue University, ECE 264 – Advanced C Programming

Stony Brook University, CSE 392 – Programming challenges, CSE 373/548 - Analysis of Algorithms

Date: 22.02.2013

HEAD OF DEPARTMENT

COURSE INSTRUCTOR,