SYLLABUS

Discipline name	Algorithms and Programming Techniques
Profile	Electronics and Telecommunications Engineering
Specialization	Telecommunications Technologies and Systems
Code	51321209
Course leader	Professor Mircea VAIDA, Ph.D. – mircea.vaida@com.utcluj.ro
Collaborators	Assistant Professor Cosmin Striletchi, Ph.D., cosmin.striletchi@com.utcluj.ro
Department	Communications
Faculty	Electronics, Telecommunications and Information Technology

Sem.	Type of discipline	Course	Applications			Course	Applications			Ind. study	TOTAL	Credits	Form of assessment
		[hours/week]				[hours/semester]							
			S	L	P		S	L	P				
2	**Fundamental**	**2**	**-**	**2**	**-**	**28**	**-**	**28**	**-**	**94**	**150**	**5**	**Coloq.**

Acquired competences :

Acquired skills (what the student is able to do):

-Basic concepts about algorithms and programming techniques
-Basic elements about OOP in C/C++ language
-Different programming abilities for sequential and linked data in C/C++
To develop:

- algorithms and programming techniques:
- recursive alg.
- searching and sorting alg.
- backtracking and divide et impera methods
- Object Oriented Programming-OOP:
- classes, objects
- overloading methods and operators
- inheritance, virtual methods and classes
- Object I/O in C++, Files in C++
- different programming abilities for sequential and linked data in C/C++
- Linked lists and trees including stack and queue

Acquired abilities (what type of equipment/ instruments/ software the student is able to handle):

At the end the students will be able to:

- Know main facilities of an OOP IDE, VC++XE
- To execute, test an debug OO applications with complex algorithms in C/C++

Prerequisites (if necessary):

COMPUTER PROGRAMMING basic knowledge

A. Course/Lecture (course/lecture titles)	
1	Recursive programming in C/C++. Stack management. Recursive and non-recursive programming methods. Backtracking.
2	Recursive and non-recursive programming methods. Divide et impera. Sorting and searching algorithms. Simple sorting: selection, insertion, interchange. Advanced sorting: quick-sort.
3	Introduction in Object Oriented Programming, OOP. Classes, Objects, members of a class.
4	Constructors, destructors, methods calling in C++. Copy constructor, arrays of objects, visibility domain.
5	Friend class and functions in C++. Static members. Struct and union in C++. Overloading methods.
6	Overloading operators in C++.
7	Inheritance in C++. Simple and multiple inheritance.
8	Virtual classes and methods. Abstract clases.
9	I/O operations in C++. iostream library, I/O with format, I/O state, manipulators functions
10	ostream, istream si fstream classes. Overriding I/O operators. C++ files.
11	Stack, queue, sequential lists.
12	Linked lists: SLL, DLL
13	Trees: definitions, properties. Binary trees, operations
14	Theoretical evaluation

B. Applications – **Laboratory** (list of laboratories), **Seminar** (contents), **Project** (project contents)	
1	Macro functions. Inline functions. Functions with implicit parameters. Functions with a variable number of parameters.
2	Recursive functions. Recursive programming methods: divide et impera.
3	Backtracking. Searching techniques.
4	Sorting techniques.
5	Classes, objects, class members.
6	The access to a class's members
7	Constructors. Destructors. Object arrays.
8	Virtual methods and classes (optional).
9	Friend functions and classes. Static members.
10	Operators overloading.
11	Simple and multiple inheritance.
12	Input/output in C++. Redefining the I/O operators.
13	Files in C++.
14	Final evaluation.

C. Individual study (reference study contents, synthesis materials, projects, applications etc.)

1 synthesis reports
10 sets of problems (the preparation part in every laboratory)
3 sets of problems (course homework)

Individual study structure	Course study	Problem solving, laboratory, project	Applications preparation	Examination time	Additional reference study	Total no. of individual study hours
Hours	28	40	5	3	18	94

References (Textbooks, courses, laboratory manual, exercise book)

In UTC-N library

1. Vaida M., Bazele dezvoltarii aplicatiilor software in electronica si telecomunicatii, curs, litografia UTC-N, 1997
2. Mircea-Florin Vaida, Petre G. Pop, Cosmin Striletchi, Ligia Chiorean, Calin G. Login, Tehnologii avansate privind dezvoltarea aplicatiilor software in limbajul C/C++, Casa Cartii de Stiinta, 2006
3. Ligia Chiorean, Mircea-Florin Vaida, Petre G. Pop, Cosmin Striletchi, , Elemente de bază şi obiectuale privind dezvoltarea aplicaţiilor în limbajul de programare C/C++, pp. 380, UTPress, 2007/2008

Supplementary materials:

-English courses
-Lab. Support on the dedicated site, http://mercur.utcluj.ro/lab (english+romanian)

Other libraries:

1. Striletchi C., Vaida M.F., Pop G.P., Chiorean Ligia, Benta K. Iulian- Tehnologii obiectuale si algoritmi de baza privind dezvoltarea aplicatiilor in limbajul C/C++, Editura Casa Cartii de Stiinta, Cluj-Napoca, 2007
2. Vaida M.F., Pop G.P., Striletchi C., Chiorean Ligia, - Aplicaţii în limbajele C/C++ şi Java, Casa Cartii de Stiinta, Cluj-Napoca, 2004.

Final evaluation	
Evaluation method	Final exam will be composed by a theoretical test T (1 hour) and solving a problem on a computer P (1 hour). The laboratory will also be evaluated

Mark components	Laboratory (mark L): activity (40%) + homework (60%);
	Theory (mark T); Problem (mark P);

Mark computation	N= (L+T+P)/3; is calculated only if: T>4 şi P>4.

Course leader,

Professor Mircea-F. VAIDA, Ph.D.