Recommended template: "Course Description, Syllabus, Course Requirements"

COURSE SYLLABUS AND COURSE REQUIREMENTS
ACADEMIC YEAR 2022/23 SEMESTER 2

Course title	*Embedded Computer Programming I.*
Course Code	**IVB335AN**
Hours/Week: le/pr/lab	**1+2**
Credits	**4**
Degree Programme	
Study Mode	**Lecture**
Requirements	**midterm**
Teaching Period	**2022/23/2**
Prerequisites	**Programming I.**
Department(s)	**MIK-AUT**
Course Director	**Zidarics Zoltán**
Course Code	**Zidarics Zoltán**

COURSE DESCRIPTION

Use of the C programming language learned in the subject Programming 1 in group work in an automated environment. The students get to know the operation and switches of the GNU C compiler. They learn how cross-compiling to ARM processors works. They learn how to use the Make utility for automating translation and deployment. They will learn how to use the CMake and Automake utilities. They also learn how to use the Git utility for version control. All tasks are submitted through the gitlab interface. During the semester, they solve simple tasks for each topic, and at the end of the semester they receive a large independent task, which is submitted via their own Git repository.

SYLLABUS

1. GOALS AND OBJECTIVES

Implementation of the design of applications running on an embedded Linux operating system and the implementation of automatic deployment tasks

2. COURSE CONTENT

	TOPICS
LABORATORY PRACTICE	1. Git 2. Make/Automake 3. Software testing (Cunit, CMocka) 4. Software documentation (Doxygen) 5. Compound data strucures, Lists, fifo,lifo List, SLIST, TAILQ 6. Parallel programming (thread, mutex) 7. Coding standards 8. Regular expressions 9. MQTT protocol 10. Final exam

DETAILED SYLLABUS AND COURSE SCHEDULE

PRACTICE, LABORATORY PRACTICE

Week	Topic	Compulsory reading; page number (from … to …)	Required tasks (assignments, tests, etc.)	Completion date, due date
1.	Introduction			
2.	Installing Linux operating system		-	
3.	Using Linux operating system		-	
4.	Using Git		-	
5.	Make/Automake		-	
6.	Software testing (CUnit/CMocka)		-	
7.	Software documentation (Doxygen)		Exam 1	
8.	Complex data structures		-	
9.	Parallel programming		-	
10.	Coding standards		Exam 2	
11.	Regular expressions		-	
12.	MQTT protocol		-	
13.	Final exam		-	
14.	Final exam		-	
15.			Exam 3	

3. ASSESSMENT AND EVALUATION

ATTENDANCE

In accordance with the Code of Studies and Examinations of the University of Pécs, Article 45 (2) and Annex 9. (Article 3) a student may be refused a grade or qualification in the given full-time course if the number of class absences exceeds 30% of the contact hours stipulated in the course description.

Method for monitoring attendance ***(e.g.: attendance sheet / online test/ register, etc.)***

attendance sheet

ASSESSMENT

Cells of the appropriate type of requirement is to be filled out (course-units resulting in mid-term grade or examination). Cells of the other type can be deleted.

course resulting in mid-term grade (PTE TVSz 40§(3))

Mid-term assessments, performance evaluation and their ratio in the final grade

Type	Assassement	Ratio in the final grade
1. Exam 1	max 10 points	25 %
2. Exam 2	max 10 points	25 %
3. Exam 3	max 20 points	50 %

Opportunity and procedure for re-takes (PTE TVSz 47§(4))

The specific regulations for improving grades and resitting tests must be read and applied according to the general Code of Studies and Examinations. E.g.: all tests and assessment tasks can be repeated/improved at least once every semester, and the tests and home assignments can be repeated/improved at least once in the first two weeks of the examination period.

Both the Applicants and the ZH, individually a minimum 40% ZH result and unexcused absences not exceeding the TVSZ, which, taking into account the topic, max. 3.

Grade calculation as a percentage

Course grade	Performance in %
excellent (5)	85 % …
good (4)	70 % ... 85 %
satisfactory (3)	55 % ... 70 %
pass (2)	40 % ... 55 %
fail (1)	Below 40 %

The lower limit given at each grade belongs to that grade.

4. SPECIFIED LITERATURE

In order of relevance. (In Neptun ES: Instruction/Subject/Subject details/Syllabus/Literature)

COMPULSORY READING AND AVAILABILITY

1. Git reference manual
2. Gitlab
3. GNU Make
4. Automake
5. CUnit
6. CMocka
7. Sys/Queue
8. Parallel processing
9. C coding standard
10. Regular expressions
11. MQTT

RECOMMENDED LITERATURE AND AVAILABILITY

1. Course syllabus