COURSE SYLLABUS AND COURSE REQUIREMENTS
ACADEMIC YEAR 2023/24 SEMESTER II.

Course title	*Engineering Programming*
Course Code	MSM617ANEG
Hours/Week: le/pr/lab	0/0/4
Credits	4
Degree Programme	Biomedical Engineering MSc
Study Mode	*full-time*
Requirements	exam
Teaching Period	spring
Prerequisites	-
Department(s)	Dept. of Technical Informatics
Course Director	Dr. Sári Zoltán
Teaching Staff	*Dr. Sári Zoltán*

COURSE DESCRIPTION

A short description of the course (max. 10 sentences).
Neptun: Instruction/Subjects/Subject Details/Basic data/Subject description

The course starts with the fundamentals of programming in general, then throughout the course the students work with different, high level, and widely used programming environments, such as LabView, Matlab, and Python. While gaining a solid knowledge of the foundations of programming the students learn about basic control structures, details of implementations, various data structures, the concept of modularity, troubleshooting and debugging techniques, resource management, files and I/O, event driven programming, simple design patterns, basic data processing and pre-processing techniques, and fundamentals of signal and image processing.

The lecture presentations give the required theoretical background supported with numerous demonstrations and examples. During the practice sessions the students solve problems of increasing difficulty to facilitate creative, and individual work on projects connected to real-world engineering problems.

SYLLABUS

Neptun: Instruction/Subjects/Subject Details/Syllabus

1. GOALS AND OBJECTIVES

Goals, student learning outcome.
Neptun: Instruction/Subjects/Subject Details/Syllabus/Goal of Instruction

The course introduces fundamental concepts of designing and developing software supported solutions for typical problems arising in engineering practice, processing, and analysis of data from various sources. The main aim of the course is to build a solid knowledge of the programming paradigms and technics, which can be applied in various programming languages and environments. After taking the course, students will be able to design and implement their own software solutions connected to real-world engineering problems.

2. COURSE CONTENT

Neptun: Instruction/Subjects/Subject Details/Syllabus/Subject content

	TOPICS
LECTURE	*1. Introduction to programming, Navigating LabView, Getting familiar with LabView UI, very simple programs using arithmetic operations* *2. Troubleshooting, debugging, Implementing a VI, Basic concepts, fundamentals of algorithms* *3. Implementing a VI, Fundamental control structures, Simple programs using sequence, decision and loops* *4. Modularity, Data structures, Algorithms working on array-like structures* *5. Resource management, File I/O, Basic data acquisition, data processing, and basic I/O*

6. *State Machines and Variables, Application of the State Machine architecture and variables*
7. *Introduction to Python, Fundamentals of Python programming, procedural programming, Basic visualization techniques, NumPy and Matplotlib,*
8. *Data processing in Python, Introduction to Pandas, File I/O*
9. *Preprocessing and filtering, data cleaning, plotting and visualization*
10. *Introduction to Matlab, Using the Matlab environment, simple expressions, data types, basic I/O, Basic visualization techniques*
11. *Fundamentals of Scripting, Basic control structures in Matlab, m-files and Functions, scope of variables, Incremental development, Debugging and Testing*
12. *Visualization, 2D and 3D plots, Fundamentals of signal and image processing*

LABORATORY PRACTICE

1. *Introduction to programming, Navigating LabView, Getting familiar with LabView UI, very simple programs using arithmetic operations*
2. *Troubleshooting, debugging, Implementing a VI, Basic concepts, fundamentals of algorithms*
3. *Implementing a VI, Fundamental control structures, Simple programs using sequence, decision and loops*
4. *Modularity, Data structures, Algorithms working on array-like structures*
5. *Resource management, File I/O, Basic data acquisition, data processing, and basic I/O*

6. *State Machines and Variables, Application of the State Machine architecture and variables*
7. *Introduction to Python, Fundamentals of Python programming, procedural programming, Basic visualization techniques, NumPy and Matplotlib,*
8. *Data processing in Python, Introduction to Pandas, File I/O*
9. *Preprocessing and filtering, data cleaning, plotting and visualization*
10. *Introduction to Matlab, Using the Matlab environment, simple expressions, data types, basic I/O, Basic visualization techniques*
11. *Fundamentals of Scripting, Basic control structures in Matlab, m-files and Functions, scope of variables, Incremental development, Debugging and Testing*
12. *Visualization, 2D and 3D plots, Fundamentals of signal and image processing*

DETAILED SYLLABUS AND COURSE SCHEDULE

ACADEMIC HOLIDAYS INCLUDED

LECTURE

week	**Topic**	**Compulsory reading; page number (from … to …)**	**Required tasks (assignments, tests, etc.)**	**Completion date, due date**
1.	Introduction to programming, Navigating LabView, Getting familiar with LabView UI, very simple programs using arithmetic operations	[1.] Lesson 1.		
2.	Troubleshooting, debugging, Implementing a VI, Basic concepts, fundamentals of algorithms	[1.] Lesson 2.		
3.	Implementing a VI, Fundamental control structures, Simple programs using sequence, decision and loops	[1.] Lesson 3.		
4.	Modularity, Data structures, Algorithms working on array-like structures	[1.] Lesson 4.		
5.	Resource management, File I/O, Basic data acquisition, data processing, and basic I/O	[1.] Lesson 5-6.		
6.	State Machines and Variables, Application of the State Machine architecture and variables	[1.] Lesson 7.		
7.			**Midterm Exam 1.**	
8.	Introduction to Python, Fundamentals of Python programming, procedural programming, Basic visualization techniques, NumPy and Matplotlib	[2.] Chapter 2-4.		

9.	Data processing in Python, Introduction to Pandas, File I/O, Preprocessing and filtering, data cleaning, plotting and visualization	[2.] Chapter 5-6. [2.] Chapter 7-9.		
10.	**Autumn brake**			
11.	Introduction to Matlab, Using the Matlab environment, simple expressions, data types, basic I/O, Basic visualization techniques	[3.] Chapter 1-2.		
12.	Fundamentals of Scripting, Basic control structures in Matlab, m-files and Functions, scope of variables, Incremental development, Debugging and Testing	[3.] Chapter 3-4.		
13.	Visualization, 2D and 3D plots, Fundamentals of signal and image processing	[3.] Chapter 8.		
14.			**Midterm Exam 2.**	
15.			**Retakes**	

PRACTICE, LABORATORY PRACTICE

week	**Topic**	**Compulsory reading; page number (from … to …)**	**Required tasks (assignments, tests, etc.)**	**Completion date, due date**
1.	Introduction to programming, Navigating LabView, Getting familiar with LabView UI, very simple programs using arithmetic operations	[1.] Lesson 1.		
2.	Troubleshooting, debugging, Implementing a VI, Basic concepts, fundamentals of algorithms	[1.] Lesson 2.		
3.	Implementing a VI, Fundamental control structures, Simple programs using sequence, decision and loops	[1.] Lesson 3.		
4.	Modularity, Data structures, Algorithms working on array-like structures	[1.] Lesson 4.		
5.	Resource management, File I/O, Basic data acquisition, data processing, and basic I/O	[1.] Lesson 5-6.		
6.	State Machines and Variables, Application of the State Machine architecture and variables	[1.] Lesson 7.		
7.			**Midterm Exam 1.**	
8.	Introduction to Python, Fundamentals of Python programming, procedural programming, Basic visualization techniques, NumPy and Matplotlib	[2.] Chapter 2-4.		
9.	Data processing in Python, Introduction to Pandas, File I/O, Preprocessing and filtering, data cleaning, plotting and visualization	[2.] Chapter 5-6. [2.] Chapter 7-9.		
10.	**Autumn brake**			
11.	Introduction to Matlab, Using the Matlab environment, simple expressions, data types, basic I/O, Basic visualization techniques	[3.] Chapter 1-2.		
12.	Fundamentals of Scripting, Basic control structures in Matlab, m-files and Functions, scope of variables, Incremental development, Debugging and Testing	[3.] Chapter 3-4.		
13.	Visualization, 2D and 3D plots, Fundamentals of signal and image processing	[3.] Chapter 8.		
14.			**Midterm Exam 2.**	
15.			**Retakes**	

3. ASSESSMENT AND EVALUATION

(Neptun: Instruction/Subjects/Subject Details/Syllabus/Examination and Evaluation System)

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 on practice sessions. Maximum allowed absence: 30% of practice classes.

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-unit with final examination

Mid-term assessments, performance evaluation and their weighting as a pre-requisite for taking the final exam

(The samples in the table to be deleted.)

Type	Assessment	*Weighting as a proportion of the pre-requisite for taking the exam*
1. *Midterm Test 1*	*max. 100%*	*50%*
2. *Midterm Test 2*	*max. 100%*	*50%*

Requirements for the end-of-semester signature

(Eg.: mid-term assessment of 40%)

Midterm performance >= 40%.

Re-takes for the end-of-semester signature (PTE TVSz 50§(2))

The specific regulations for grade betterment and re-take must be read and applied according to the general Code of Studies and Examinations. E.g.: all the tests and the records to be submitted can be repeated/improved each 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.

Each midterm test can be retaken one time during the semester.

Type of examination *(written, oral): written*

The exam is successful if the result is minimum 40 %. *(The minimum cannot exceed 40%.)*

Calculation of the grade *(TVSz 47§ (3))*

The mid-term performance accounts for ***50*** %, the performance at the exam accounts for ***50*** % in the calculation of the final grade.

Calculation of the final grade based on aggregate performance in 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.] LabView Core 1 Course Manual, National Instruments, 2012
[2.] Wes McKinney, Pyhon for Data Analysis, O'Reilly, 2018
[3.] Andrew P. King, Paul Aljabar, Matlab programming for Biomedical Engineers and scientists, Elsevier Academic Press, 2017