SYLLABUS

Affective Computing

University year 2025-2026

1. Information regarding the programme

1.1. Higher education institution	Babes-Bolyai University, Cluj-Napoca
1.2. Faculty	Faculty of Mathematics and Computer Science
1.3. Department	Department of Computer Science
1.4. Field of study	Computer Science
1.5. Study cycle	Bachelor
1.6. Study programme/Qualification	Computer Science – English line
1.7. Form of education	

2. Information regarding the discipline

2.1. Name of the discipline	**Affective Computing**					Discipline code	**MLE5150**
2.2. Course coordinator	Lecturer dr. eng. Iulian Bența						
2.3. Seminar coordinator	Lecturer dr. eng. Iulian Bența						
2.4. Year of study	3	2.5. Semester	5	2.6. Type of evaluation	C	2.7. Discipline regime	Optional

3. Total estimated time (hours/semester of didactic activities)

3.1. Hours per week	**3**	of which: 3.2 course	**2**	3.3 seminar/laboratory/project	**3**
3.4. Total hours in the curriculum	70	of which: 3.5 course	28	3.6. seminar/ laboratory/ project	42
Time allotment for individual study (ID) and self-study activities (SA)					**hours**
Learning using manual, course support, bibliography, course notes (SA)					3
Additional documentation (in libraries, on electronic platforms, field documentation)					12
Preparation for seminars/labs, homework, papers, portfolios and essays					12
Tutorship					1
Evaluations					2
Other activities:					
3.7. Total individual study hours	**30**				
3.8. Total hours per semester	**100**				
3.9. Number of ECTS credits	**4**				

4. Prerequisites (if necessary)

4.1. curriculum	- Algorithms, Data structures
4.2. competencies	- High level programming language (OOP) skills

5. Conditions (if necessary)

5.1. for the course	A room with Internet access and presentation devices
5.2. for the seminar /lab activities	A room with computers (with up-to-date processing power, minimum 16 GB RAM) and high-speed Internet access

6.1. Specific competencies acquired [1]

[1] One can choose either competences or learning outcomes, or both. If only one option is chosen, the row related to the other option will be deleted, and the kept one will be numbered 6.

Professional/ essential competencies	• advanced programming skills in high-level programming languages • development and maintenance of software systems • use of software tools in an interdisciplinary context • use of theoretical foundations of computer science as well as of formal models • design and management of databases • design and administration of computer networks • use of artificial intelligence concepts and techniques to solve real-world problems
Transversal competencies	• application of organized and efficient work rules, of responsible attitudes towards the didactic-scientific field, to bring creative value to own potential, with respect for professional ethics principles and norms • efficient development of organized activities in an interdisciplinary group and the development of empathetic abilities for interpersonal communications, to relate to and cooperate with various groups • use of efficient methods and techniques to learn, inform, research and develop the abilities to bring value to knowledge, to adapt at the requirements of a dynamical society and to communicate efficiently in Romanian language and in an international language.

6.2. Learning outcomes

Knowledge	- The student has the knowledge necessary for using computers, developing programs and software applications, information processing. - The student has knowledge related to programming, math, engineering and technology and has the necessary skills to use them in the creation of complex computer systems. - The student has the knowledge necessary to design, analyze and manage databases. - The student has adequate knowledge related to the use of integrated development environments for the purpose of creating large complex applications. - The student has knowledge of multiple programming languages and is able to write applications in compiled, interpreted or dynamic languages with the ability to choose the appropriate programming language for the specific application to be developed. - The student has the knowledge necessary to select and use appropriate training procedures to facilitate the assimilation of knowledge. - The student has the knowledge necessary to review the literature. - The student has the knowledge necessary to process and verify data and information. - The student has the necessary knowledge of software life cycle stages and software process models. - The student knows the concepts related to software modeling and is able to implement functional and non-functional requirements described in specific documents for the analysis and design of software systems. - The student has the knowledge to apply model-based software development techniques. - The student has the necessary knowledge related to the UML language as well as the ability to use CASE tools to understand, document and implement software systems. - The student knows the basic aspects of software management. - The student is familiar with traditional and agile development methodologies. - The student knows the methods of testing and verification of software systems. - The student has knowledge of the basics of operating system-specific programming, and has fundamental knowledge of scripting programming. - The student is familiar with tools used to test, debug, validate software applications. - The student is familiar with project management tools, version control systems, and continuous integration/continuous delivery (CI/CD) concepts, methods, tools.

Skills	The student is able to: - The student has the skills necessary to design computer programs and analyze software systems. - The student is able to combine diverse information to formulate solutions and generate development ideas for new products and applications. - The student is able to apply architectural templates, design templates and industry best practices to design software applications of high complexity. - The student has the ability to evaluate different possible architectures and solutions for a problem and choose the appropriate one for the specific requirements and constraints of the application to be developed. - The student has the necessary skills to understand and use object-oriented programming concepts to develop software applications of medium complexity. - The student has the ability to understand and use design templates for application development. - The student has the necessary skills to develop applications with graphical user interfaces using architectural templates appropriate to the specifics of applications with user interaction. - The student has the ability to choose and use existing modules and environments for application development. - The student has the ability to understand and communicate information effectively. The student is able to present and explain methods, algorithms, paradigms and techniques used in different branches of computer science. - The student is able to define/identify/understand research problems in computer science. - The student is familiar with international academic research databases and digital libraries (Web of Science, ACM Digital Library, IEEE Xplore, Springer, Elsevier, CiteSeerX, etc.). - The student has the necessary skills to apply different methods and tools to visualize research results. - The student is able to write a scientific report. - The student has the ability to observe and obtain information from various sources. - The student has the necessary skills to install and configure operating systems.
Responsibility and autonomy:	- The student has the ability to choose and use programming paradigms (procedural, object-oriented, functional) for the realization of software applications appropriate to the specific domain of the developed application. - The student has the ability to develop, design and create new applications, systems or products using best practices in the field. - The student has the ability to identify complex problems and examine related issues to develop solution options and implement solutions. - The student has the ability to identify the educational needs of others and develop educational or training and development programs. - The student has the ability to conduct research in educational sciences. - The student has the ability to identify and use appropriate tools to support learning-teaching. - The student has the ability to introduce new, innovative elements into the instructional-educational process if deemed useful or necessary. - The student has the ability to apply general rules to specific problems and produce relevant solutions. - The student has the ability to use research support tools.

7. Objectives of the discipline (outcome of the acquired competencies)

7.1 General objective of the discipline	• Developing the ability to analyze, design and implement user's affective states adapted applications
7.2 Specific objective of the discipline	• Acquaintance with signals and algorithms for mono, bi and multimodal affective states • Skills to develop complex modular applications with signal processing, feature extraction and machine learning

8. Content

8.1 Course	**Teaching methods**	**Remarks**
1. Introduction to Affective Computing (examples, historical facts, definitions)	Presentation, interactive lecture, discussions, case studies, problem solving	2 hours
2. Affect Models (Russell, activation-valence, OCC, appraisal)	Presentation, interactive lecture, discussions, case studies, problem solving	2 hours

3. Affective States Representation (discrete, dimensional, fuzzy; measures in modelling)	Presentation, interactive lecture, discussions, case studies, problem solving	2 hours
4. Facial Expression Recognition (models, approaches, model fusion, deep learning)	Presentation, interactive lecture, discussions, case studies, problem solving	2 hours
5. Voice-based Affective States Assessment (feature extraction, pattern recognition)	Presentation, interactive lecture, discussions, case studies, problem solving	2 hours
6. Physiological Affective States Detection (feature extraction, pattern recognition)	Presentation, interactive lecture, discussions, case studies, problem solving	2 hours
7. Affective States Assessment from Other Communication Channels (kinesthetic-postural, contextual, text content)	Presentation, interactive lecture, discussions, case studies, problem solving	2 hours
8. Multimodal Affective States Detection (sensor fusion, computing infrastucture)	Presentation, interactive lecture, discussions, case studies, problem solving	2 hours
9. Presentation and discussion of the Theoretical Projects	Presentation, interactive lecture, discussions, case studies, problem solving	2 hours
10. Ethical Aspects in Affective Computing	Presentation, interactive lecture, discussions, case studies, problem solving	2 hours
11 – 12. Presentation and discussion of the Practical Projects (I and II)	Presentation, interactive lecture, discussions, case studies, problem solving	4 hours
13-14. Research Challenges in Affective Computing (I and II)	Presentation, interactive lecture, discussions, case studies, problem solving	4 hours

Bibliography:

1. Emotionale Intelligenz erhöhen: Emotionen wahrnehmen, verstehen und ausdrücken, by Casten Voller, ISBN-13: 978-1521902776, ISBN-10: 1521902771, 2017

2. Mensch und Maschine: Wie künstliche Intelligenz und Roboter unser Leben verändern, by Thomas Ramge (Author), Dinara Galieva (Illustrator), ISBN-13: 978-3150194997, ISBN-10: 3150194997, 2018

3. The Oxford Handbook of Affective Computing (Oxford Library of Psychology) 1st Edition, by Rafael A. Calvo (Editor), Sidney D'Mello (Editor), Jonathan Gratch (Editor), Arvid Kappas (Editor), ISBN-13: 978-0199942237, ISBN-10: 9780199942237, 2014

4. Emotions and Affect in Human Factors and Human-Computer Interaction, by Myounghoon Jeon (Editor), ISBN-13: 978-0128018514, ISBN-10: 0128018518, 2017.

5. Deep Learning. Das umfassende Handbuch: Grundlagen, aktuelle Verfahren und Algorithmen, neue Forschungsansätze, Ian Goodfellow, Yoshua Bengio, Aaron Courville, mitp Professional, 2018

8.2 Seminar / laboratory	**Teaching methods**	**Remarks**
1. Project themes presentation. Project analysis and design phase.	Explanations, Demonstrations, Discussion, Brainstorming, Case studies, Collaboration	2 hours
2. Hands-on experience with available Affective Computing solutions	Explanations, Demonstrations, Discussion, Brainstorming, Case studies, Collaboration	2 hours
3. Designing and implementing a simple Facial Expression Recognition System	Explanations, Demonstrations, Discussion, Brainstorming, Case studies, Collaboration	2 hours
4. Designing and implementing a bimodal Affective State Assessment System	Explanations, Demonstrations, Discussion, Brainstorming, Case studies, Collaboration	2 hours

5. Using Mobile and Wearable Devices for Affective Computing	Explanations, Demonstrations, Discussion, Brainstorming, Case studies, Collaboration	2 hours
6-7. Development and refinement of the Practical Projects (I and II)	Explanations, Demonstrations, Discussion, Brainstorming, Case studies, Collaboration	4 hours

Bibliography:

1. Emotionale Intelligenz erhöhen: Emotionen wahrnehmen, verstehen und ausdrücken, by Casten Voller, ISBN-13: 978-1521902776, ISBN-10: 1521902771, 2017

2. Mensch und Maschine: Wie künstliche Intelligenz und Roboter unser Leben verändern, by Thomas Ramge (Author), Dinara Galieva (Illustrator), ISBN-13: 978-3150194997, ISBN-10: 3150194997, 2018

3. The Oxford Handbook of Affective Computing (Oxford Library of Psychology) 1st Edition, by Rafael A. Calvo (Editor), Sidney D'Mello (Editor), Jonathan Gratch (Editor), Arvid Kappas (Editor), ISBN-13: 978-0199942237, ISBN-10: 9780199942237, 2014

4. Emotions and Affect in Human Factors and Human-Computer Interaction, by Myounghoon Jeon (Editor), ISBN-13: 978-0128018514, ISBN-10: 0128018518, 2017.

5. Deep Learning. Das umfassende Handbuch: Grundlagen, aktuelle Verfahren und Algorithmen, neue Forschungsansätze, Ian Goodfellow, Yoshua Bengio, Aaron Courville, mitp Professional, 2018.

9. Corroborating the content of the discipline with the expectations of the epistemic community, professional associations and representative employers within the field of the program

The curriculum of this course aligns to the guidelines of ACM and IEEE

The software organizations recognize the importance of the concepts discussed during this course for the development of functional, user-friendly and intelligent products.

10. Evaluation

Activity type	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Percentage of final grade
10.4 Course	- Basic knowledge of the Affective Computing domain - Operationalization of the principles and technologies to design and develop affective states assessment applications	Theoretical Projects Presentation	30%
10.5 Seminar/laboratory	- Analyze, Design, Implementation and Testing of affective states assessment applications	Practical Projects Presentation	50%
		Systematical observation of the student through the laboratory activities	20%

10.6 Minimum standard of performance

Each student should demonstrate that he/she reached an acceptable level of knowledge and understanding of the Affective Computing domain, that she/he is able to express the knowledge in a coherent form and that is able to practically apply those in order to solve real world problems for the user benefit in an ethical manner.

It is necessary to obtain a minimum grade of 5 (average of Course and Laboratory) and to demonstrate a minimal but functional and original affective assessment application in order to pass this discipline.

11. Labels ODD (Sustainable Development Goals)[2]

Not applicable.

Date: April 2025	Signature of course coordinator Lecturer dr. eng. Iulian Benţa	Signature of seminar coordinator Lecturer dr. eng. Iulian Benţa

Date of approval: April 2025	Signature of the head of department Assoc. Prof. Ph.D. Adrian STERCA

[2] Keep only the labels that, according to the *Procedure for applying ODD labels in the academic process*, suit the discipline and delete the others, including the general one for *Sustainable Development* – if not applicable. If no label describes the discipline, delete them all and write „*Not applicable*.”.