SYLLABUS

1. Information regarding the programme

1.1 Higher education institution	**Babeş-Bolyai University**
1.2 Faculty	**Faculty of Mathematics and Computer Science**
1.3 Department	**Department of Computer Science**
1.4 Field of study	**Computers and Information Technology**
1.5 Study cycle	**Bachelor**
1.6 Study programme / Qualification	**Information Engineering**

2. Information regarding the discipline

2.1 Name of the discipline (en) / (ro)			**Computer Science Investigations : IoT** / Investigații în Știința Calculatoarelor : IoT				
2.2 Course coordinator			**Lect. Dr. Mircea Ioan-Gabriel**				
2.3 Seminar coordinator			**Lect. Dr. Mircea Ioan-Gabriel**				
2.4. Year of study	**4**	2.5 Semester	**8**	2.6. Type of evaluation	**C**	2.7 Type of discipline	**Optional DS**
2.8 Code of the discipline							

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

3.1 Hours per week	5	Of which: 3.2 course	2	3.3 seminar/laboratory	1LP 2P
3.4 Total hours in the curriculum	70	Of which: 3.5 course	28	3.6 seminar/laboratory	42

Time allotment:	hours
Learning using manual, course support, bibliography, course notes	11
Additional documentation (in libraries, on electronic platforms, field documentation)	11
Preparation for seminars/labs, homework, papers, portfolios and essays	11
Tutorship	11
Evaluations	6
Other activities:	5

3.7 Total individual study hours	55
3.8 Total hours per semester	125
3.9 Number of ECTS credits	5

4. Prerequisites (if necessary)

4.1. curriculum	•
4.2. competencies	•

5. Conditions (if necessary)

5.1. for the course	•
5.2. for the seminar / lab activities	•

6. Specific competencies acquired

Professional competencies	C2.1 Describing the structure and operation of hardware, software and communication components C2.2 Explaining the role, interaction and operation of hardware, software and communication components C2.3 Construction of hardware and software components of computing systems using design methods, languages, algorithms, data structures, protocols and technologies C2.4 Metric based evaluation of functional and non-functional characteristics of computing systems C2.5 Implementation of hardware, software and communication components C3.1 Identifying classes of problems and solving methods that are specific to computing systems C3.2 Using interdisciplinary knowledge, solution patterns and tools, making experiments and interpreting their results C3.3 Applying solution patterns using specific engineering tools and mehods C3.4 Comparatively and experimentaly evaluation of the alternative solutions for performance optimization C3.5 Developing and implementing information system solutions for concrete problems
Transversal competencies	CT1 Honorable, responsible, ethical behavior, in the spirit of the law, to ensure the professional reputation CT3 Demonstrating initiative and pro-active behavior for updating professional, economical and organizational culture knowledge

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

7.1 General objective of the discipline	• The theoretical and practical training of students for thorough computer science investigations in the realm of IoT
7.2 Specific objective of the discipline	• Understanding and properly applying • scientific research methodology • IoT Development methodology • Software Development practices • Machine Learning practices • Algorithmic Reasoning • Testing practices

8. Content

8.1 Course	Teaching methods	Remarks
1. Investigating real-life problems. The social purpose of Computer Science. Ethics and morality in Software Development and Science	Interactive exposure Explanation Conversation Didactical demonstration	
2. Evaluating the quality of scientific papers and publications. Establishing a proper bibliographic basis	Interactive exposure Explanation Conversation Didactical demonstration	
3. Qualitatively sorting and filtering bibliographic resources and assessing the state-of-the-art of the problem domain. Clustering the bibliography based on the main research directions	Interactive exposure Explanation Conversation Didactical demonstration	
4. The danger of plagiarism and ways of combating it. Identifying unexplored or improvable research niches	Interactive exposure Explanation Conversation Didactical demonstration	
5. Harvesting datasets from the state-of-the-art of the research niche as benchmarks for performance evaluation	Interactive exposure Explanation Conversation Didactical demonstration	
6. Artificial Intelligence vs. Classical Algorithmics: choosing the right original approach	Interactive exposure Explanation Conversation Didactical demonstration	
7. Designing and Developing an API for the proposed approach. Programming principles and good practices. Choosing the architecture, design patterns, language and technology. API testing	Interactive exposure Explanation Conversation Didactical demonstration	
8. Data analysis, preprocessing and visualisation for algorithm training and performance evaluation. Training, Validation, Testing. Performance assessment	Interactive exposure Explanation Conversation Didactical demonstration	
9. Optimisation via hyper-parametrization	Interactive exposure Explanation Conversation Didactical demonstration	
10. Software design aspects: design patterns and principles	Interactive exposure Explanation	

	Conversation Didactical demonstration	
11.Software development aspects: cybersecurity, UX, testing	Interactive exposure Explanation Conversation Didactical demonstration	
12. IoT: Background and Challenges	Interactive exposure Explanation Conversation Didactical demonstration	
13. IoT design and development aspects	Interactive exposure Explanation Conversation Didactical demonstration	
14. Building and deploying an IoT-enhanced software product	Interactive exposure Explanation Conversation Didactical demonstration	

Bibliography

Programming Fundamentals

1. Donald E. Knuth. 2011. The Art of Computer Programming: Combinatorial Algorithms, Part 1 (1st. ed.). Addison-Wesley Professional.
2. Brian W. Kernighan and Dennis M. Ritchie. 1988. The C Programming Language (2nd. ed.). Prentice Hall Professional Technical Reference.
3. Bruce Eckel. 2000. Thinking in C++, Volume I: Introduction to Standard C++, Second Edition (2nd. ed.). Prentice Hall PTR, USA.
4. Dijkstra, Edsger W. A Discipline of Programming. 1976.
5. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2009. Introduction to Algorithms, Third Edition (3rd. ed.). The MIT Press.
6. Thomas H. Cormen. 2013. Algorithms Unlocked. The MIT Press.
7. 1999. Refactoring: improving the design of existing code. Addison-Wesley Longman Publishing Co., Inc., USA.
8. Antti Laaksonen, Guide to Competitive Programming - Learning and Improving Algorithms Through Contests, Second Edition. Undergraduate Topics in Computer Science, Springer 2020, ISBN 978-3-030-39356-4, pp. 1-296

Software Design and Architecture

1. Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. 1995. Design patterns: elements of reusable object-oriented software. Addison-Wesley Longman Publishing Co., Inc., USA.
2. Robert C. Martin. 2008. Clean Code: A Handbook of Agile Software Craftsmanship (1st. ed.). Prentice Hall PTR, USA.
3. Robert C. Martin. 2017. Clean Architecture: A Craftsman's Guide to Software Structure and Design (1st. ed.). Prentice Hall Press, USA.
4. Bhuvan Unhelkar. 2017. Software Engineering with UML (1st. ed.). Auerbach Publications, USA.
5. Alan Dennis, Barbara Haley Wixom, and David Tegarden. 2015. Systems Analysis and

Design: An Object-Oriented Approach with UML (5th. ed.). Wiley Publishing.
6. Richards, M., Ford, N., & Safari, an O'Reilly Media Company. (2020). Fundamentals of Software Architecture.
7. Sam Newman. 2015. Building Microservices (1st. ed.). O'Reilly Media, Inc.
8. Irakli Nadareishvili, Ronnie Mitra, Matt McLarty, and Mike Amundsen. 2016. Microservice Architecture: Aligning Principles, Practices, and Culture (1st. ed.). O'Reilly Media, Inc.

Data Persistency

1. Thomas Erl, Ricardo Puttini, and Zaigham Mahmood. 2013. Cloud Computing: Concepts, Technology & Architecture (1st. ed.). Prentice Hall Press, USA.
2. Kief Morris. 2016. Infrastructure as Code: Managing Servers in the Cloud (1st. ed.). O'Reilly Media, Inc.
3. Kavis, Michael J. Architecting the Cloud: Design Decisions for Cloud Computing Service Models (saas, Paas, and Iaas). New York: John Wiley & Sons, 2014.

Artificial Intelligence

1. Stuart Russell and Peter Norvig. 2009. Artificial Intelligence: A Modern Approach (3rd. ed.). Prentice Hall Press, USA.
2. GeÌron, AureìlienÌ. Hands-on Machine Learning with Scikit-Learn, Keras and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. 2nd ed., O'Reilly, 2019.
3. David James. 2018. Introduction to Machine Learning with Python: A Guide for Beginners in Data Science (1st. ed.). CreateSpace Independent Publishing Platform, North Charleston, SC, USA.
4. Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. Deep Learning. The MIT Press.

Computer Networking and Security

1. Andrew Tanenbaum. 2002. Computer Networks (4th. ed.). Prentice Hall Professional Technical Reference.
2. Jon Erickson. 2008. Hacking: the art of exploitation, 2nd edition (Second. ed.). No Starch Press, USA.
3. Kevin D. Mitnick and William L. Simon. 2003. The Art of Deception: Controlling the Human Element of Security. John Wiley & Sons, Inc., USA.
4. Chwan-Hwa (John) Wu and J. David Irwin. 2013. Introduction to Computer Networks and Cybersecurity (1st. ed.). CRC Press, Inc., USA.
5. James Graham, Ryan Olson, and Rick Howard. 2010. Cyber Security Essentials (1st. ed.). Auerbach Publications, USA.
6. Mitnick, Kevin D., and William L. Simon. Ghost in the Wires: My Adventures As the World's Most Wanted Hacker. New York: Little, Brown and Company, 2011.

Testing

1. Cem Kaner, Jack L. Falk, and Hung Quoc Nguyen. 1999. Testing Computer Software, Second Edition (2nd. ed.). John Wiley & Sons, Inc., USA.
2. Michael Sutton, Adam Greene, and Pedram Amini. 2007. Fuzzing: Brute Force Vulnerability Discovery. Addison-Wesley Professional.
3. Hsu, Tony Hsiang-Chih. Practical Security Automation and Testing: Tools and Techniques for Automated Security Scanning and Testing in DevSecOps. Birmingham: Packt Publishing Ltd, 2019.

IoT

1. Robert Oshana and Mark Kraeling. 2013. Software Engineering for Embedded Systems: Methods, Practical Techniques, and Applications (1st. ed.). Newnes, USA.
2. Naveen Balani and Rajeev Hathi. 2015. Enterprise IoT: A Definitive Handbook. CreateSpace Independent Publishing Platform, North Charleston, SC, USA.
3. Vuppalapati, C.,Building Enterprise IoT Applications,CRC Press/Taylor & Francis

Group,2019
4. Dimitrios Serpanos and Marilyn Wolf. 2017. Internet-of-Things (IoT) Systems: Architectures, Algorithms, Methodologies (1st. ed.). Springer Publishing Company, Incorporated.
5. Samuel Greengard. 2015. The Internet of Things. The MIT Press.

Scientific Research

1. Justin Zobel. 2015. Writing for Computer Science (3rd. ed.). Springer Publishing Company, Incorporated.
2. Philip W.L. Fong. 2009. Reading a computer science research paper. SIGCSE Bull. 41, 2 (June 2009), 138–140. DOI:https://doi.org/10.1145/1595453.1595493
3. Lury, Celia. Routledge Handbook of Interdisciplinary Research Methods. , 2018.
4. Repko, Allen F, et al. Case Studies in Interdisciplinary Research. Thousand Oaks, CA: SAGE Publications, Inc., 2012. SAGE Research Methods. 13 Jan 2021, doi: http://www.doi.org/10.4135/9781483349541
5. Repko, Allen F, Rick Szostak, and Michelle P. Buchberger. Introduction to Interdisciplinary Studies. , 2017.
6. Repko, Allen F, and Rick Szostak. Interdisciplinary Research: Process and Theory. , 2017.

8.2 Project	Teaching methods	Remarks
Week 1: How to make a sentient AI Robot Week 2: IoT Fundamentals: sensors Week 3: IoT Fundamentals: Spatial Movement Week 4: IoT Fundamentals: Electronics and Electric Engines Week 5: Supervised Learning Week 6:Unsupervised Learning Week 7:Reinforcement Learning Week 8:Spiking Neural Networks Week 9:Face Detection Week 10:Face Recognition Week 11:Sound Processing(Speech to text/Text to speech) Week 12:Emotion Detection Week 13:Music and Image Generation Week 14:Putting it all together	Lab assignment Explanation Conversation Scientific method	
8.2 Laboratory	Teaching methods	Remarks
Lab 1-2 The student will gradually write, based on the selected bibliography following the scientific investigation process, the chapters State-of-the-art and Proposed Solution of a future scientific article on the topic. At the same time the student will draw out the algorithm and its implementation with the help of the most suitable programming languages and technologies.	Lab assignment Explanation Conversation Scientific method	

Lab 3 The student will test the performance of the proposed solution on benchmark datasets from the literature The scientific article will be completed with the chapters Performance Evaluation, Conclusions and future development, and last but not least, Introduction	Lab assignment Explanation Conversation Scientific method	
Lab 4 The student will have two deliverables completed: - the API - specified, tested and documented (preferably on git) - the scientific article describing the whole process that led to the emergence of the API and the evaluation of its performance These deliverables will be presented in the last week of school before the Christmas holidays in front of a commission composed of representatives of partner companies in a stand-alone event (not during one of the classes). The best scientific research conducted so far will be mentored by real professionals from the industry for the rest of the semester with the purpose of embedding them in actual IoT-enhanced software products.	Lab assignment Explanation Conversation Scientific method	
Lab 5-7 Once the investigation is chosen by the mentor, during the winter holidays, the weeks after the holidays and in the exam session, the team of mentors and students will develop an industry-level IoT application and will perfect the proposed solution to achieve the final deliverable: - A software product designed and developed correctly, specified, tested and documented (accessible on git) to illustrate the utility of the proposed scientific solution in a concrete IoT context	Lab assignment Explanation Conversation Scientific method	

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 course follows the IEEE and ACM curricular recommendations for computer science studies

10. Evaluation

Type of activity	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Share in the grade (%)
10.4 Course	Proper understanding of scientific research methodologies in Computer Science	Scientific article	20%
	Proper scientific ethics		
10.5 Seminar / lab activities	Framework design and architecture. Programming principles and practices. Testing.	The proposed API - specified, tested and documented (preferably on git)	20%
	Software application design. Programming principles and practices. Testing.	A software product (accessible on git) to illustrate the utility of the proposed scientific solution	20%
	IoT software design. Programming principles and practices. Testing.	IoT Module	20%
10.5.1 Project	IoT software design. Programming principles and practices. Testing	An AI sentient robot	20%
10.6 Minimum performance standards			
• Minimum 5 grade for the course, project and lab activity			

Date	Signature of course coordinator	Signature of seminar coordinator
23.05.2022	Lect. Dr. Mircea Ioan-Gabriel	Lect. Dr. Mircea Ioan-Gabriel

Date of approval

Signature of the head of department

Prof. Dr. Diosan Laura-Silvia

24.05.2022