



	<p>Ministry of Higher Education and Scientific Research - Iraq Al-Zahrawi University College of Science Medical Physics Department</p>	
---	--	---

MODULE DESCRIPTOR FORM

Module Information			
Module Title	Computer 2	Module Delivery	
Module Type	BASIC	<input checked="" type="checkbox"/> Theory <input type="checkbox"/> Lecture <input checked="" type="checkbox"/> Lab <input type="checkbox"/> Tutorial <input type="checkbox"/> Practical <input type="checkbox"/> Seminar	
Module Code	ZU-Sc- UREQ151		
ECTS Credits	3		
SWL (hr/sem)	75		
Module Level	2		
Administering Department	Medical Physics	College	Science
Module Leader	Bahaa Turki Raji	e-mail	bahaa881212@gmail.com
Module Leader's Acad. Title	Assistant Lecturer	Module Leader's Qualification	M.Sc. Degree in Communication Engineering
Module Tutor		e-mail	
Peer Reviewer Name		e-mail	
Review Committee Approval		Version Number	

Relation With Other Modules

Prerequisite module	None	Semester	
Co-requisites module	None	Semester	

Module Aims, Learning Outcomes and Indicative Contents

Module Aims	<p>Module Aims</p> <ol style="list-style-type: none"> Develop Core Competencies: To provide students with a thorough understanding of core concepts in computer science, including algorithms, programming, data structures, and software development. Enhance Problem-Solving Skills: To equip students with the ability to solve complex problems using computational approaches and logical reasoning. Introduce Advanced Topics: To expose students to advanced topics such as artificial intelligence, machine learning, cybersecurity, and data analytics. Foster Ethical Awareness: To instill an understanding of ethical considerations and societal impacts of technology and computing. Encourage Research and Innovation: To inspire students to engage in research activities and innovative projects. <p>Learning Outcomes</p> <p>By the end of the module, students should be able to:</p> <ol style="list-style-type: none"> Understand Key Concepts: Demonstrate a deep understanding of fundamental and advanced computer science concepts. Apply Programming Skills: Develop and implement efficient algorithms and programs using multiple programming languages. Analyze and Solve Problems: Critically analyze problems and develop effective computational solutions. Conduct Research: Perform independent research and contribute to the body of knowledge in the field of computer science. Ethical Practices: Recognize and adhere to ethical standards and practices in computing and technology. Communicate Effectively: Present technical information clearly and effectively, both in written and oral forms.
--------------------	---

	<p>7. Work Collaboratively: Collaborate effectively within multidisciplinary teams.</p>
<p>Module Learning Outcomes</p>	<p>Learning Outcomes</p> <ol style="list-style-type: none"> 1. Knowledge and Understanding: <ul style="list-style-type: none"> ○ Demonstrate a comprehensive understanding of fundamental computer science principles and concepts, such as algorithms, data structures, and software engineering. ○ Explain the theoretical foundations of computer science, including computational theory and complexity. ○ Discuss the ethical, social, and legal implications of computing technologies. 2. Practical Skills: <ul style="list-style-type: none"> ○ Design and implement software solutions using appropriate programming languages and development tools. ○ Apply problem-solving techniques to develop efficient and effective algorithms. ○ Use software engineering principles and methodologies to manage and execute projects. 3. Research and Analysis: <ul style="list-style-type: none"> ○ Conduct research to stay updated with the latest advancements in computer science. ○ Analyze and evaluate computing systems and technologies. ○ Critically assess and synthesize research findings to inform practice and development. 4. Communication and Collaboration: <ul style="list-style-type: none"> ○ Communicate technical information effectively through written reports and oral presentations. ○ Work collaboratively in teams to achieve common goals and solve complex problems. ○ Demonstrate interpersonal skills and the ability to give and receive constructive feedback. 5. Professional Development: <ul style="list-style-type: none"> ○ Exhibit professionalism and ethical behavior in all aspects of work. ○ Develop a lifelong learning mindset to continuously improve skills and knowledge. ○ Prepare for professional careers in computer science by understanding industry standards and practices. 6. Application and Integration: <ul style="list-style-type: none"> ○ Apply theoretical knowledge to practical scenarios and real-world problems. ○ Integrate various computer science concepts and skills to develop comprehensive solutions.

	<ul style="list-style-type: none"> ○ Demonstrate the ability to learn and adapt to new technologies and methodologies.
<p>Indicative Contents</p>	<ul style="list-style-type: none"> ● Introduction to Computer Science <ul style="list-style-type: none"> ● History and evolution of computing ● Overview of computer systems and architecture ● Basics of computer hardware and software ● Programming Fundamentals <ul style="list-style-type: none"> ● Basic syntax and semantics of programming languages (e.g., Python, Java, C++) ● Control structures, functions, and data types ● Introduction to object-oriented programming concepts ● Data Structures and Algorithms <ul style="list-style-type: none"> ● Arrays, linked lists, stacks, queues, trees, and graphs ● Sorting and searching algorithms (e.g., bubble sort, merge sort, binary search) ● Algorithm complexity and Big-O notation ● Software Development <ul style="list-style-type: none"> ● Software development lifecycle (SDLC) ● Agile methodologies and version control (e.g., Git) ● Debugging and testing techniques ● Database Systems <ul style="list-style-type: none"> ● Relational databases and SQL ● Data modeling and database design ● Transactions and concurrency control ● Operating Systems <ul style="list-style-type: none"> ● Processes, threads, and concurrency ● Memory management and file systems ● Security and protection mechanisms ● Networks and Communications <ul style="list-style-type: none"> ● Network architectures and protocols

	<ul style="list-style-type: none"> • Data transmission and network security • Internet and web technologies • Artificial Intelligence and Machine Learning <ul style="list-style-type: none"> • Introduction to AI concepts and techniques • Machine learning algorithms and applications (e.g., supervised learning, unsupervised learning) • Neural networks and deep learning • Cybersecurity <ul style="list-style-type: none"> • Fundamentals of cybersecurity and cryptography • Threats, vulnerabilities, and risk management • Secure coding practices and incident response • Human-Computer Interaction <ul style="list-style-type: none"> • Principles of user interface design • Usability testing and evaluation • Accessibility and user experience (UX) design • Computer Graphics <ul style="list-style-type: none"> • Fundamentals of computer graphics • Rendering techniques and algorithms • Graphics programming and visualization • Theory of Computation <ul style="list-style-type: none"> • Automata theory and formal languages • Turing machines and computational complexity • Decidability and undecidability • Capstone Project <ul style="list-style-type: none"> • Integration of knowledge and skills from various areas of computer science • Collaborative project development • Presentation and demonstration of the final project
Learning and Teaching Strategies	
Strategies	

Study Strategies

1. **Understand the Basics:** Make sure you have a solid grasp of fundamental concepts like algorithms, data structures, and programming principles. These are the building blocks for more advanced topics.
2. **Practice Regularly:** Regular coding practice helps reinforce concepts and improves problem-solving skills. Websites like LeetCode, HackerRank, and Codeforces are great for practicing coding problems.
3. **Work on Projects:** Building real-world projects allows you to apply your knowledge and gain practical experience. Choose projects that interest you and align with your career goals.
4. **Learn Multiple Languages:** While specializing in one programming language is important, having a working knowledge of several languages (e.g., Python, Java, C++) can be beneficial.
5. **Participate in Competitions:** Joining coding competitions and hackathons can challenge you to think creatively and work under pressure. It's also a great way to network with other professionals.

Learning Resources

1. **Online Courses:** Platforms like Coursera, edX, and Udacity offer comprehensive courses on various computer science topics, often taught by industry experts and professors.
2. **Books:** Invest in well-regarded textbooks like "Introduction to Algorithms" by Cormen, Leiserson, Rivest, and Stein (CLRS), and "Clean Code" by Robert C. Martin.
3. **Tutorials and Documentation:** Websites like W3Schools, Mozilla Developer Network (MDN), and Stack Overflow are invaluable for tutorials and troubleshooting.
4. **Research Papers and Journals:** Stay updated with the latest advancements by reading research papers and journals in your areas of interest.

Problem-Solving Techniques

1. **Break Down Problems:** Divide complex problems into smaller, manageable parts. This approach makes it easier to tackle each part individually.
2. **Think Algorithmically:** Develop a habit of thinking about problems in terms of algorithms and data structures. This mindset will help you design efficient solutions.
3. **Debugging Skills:** Master the art of debugging. Use tools like print statements, debuggers, and logging to identify and fix issues in your code.

4. **Optimize Solutions:** Always strive to improve the efficiency of your solutions. Consider time and space complexity when designing algorithms.

Professional Development

1. **Networking:** Join professional organizations, attend conferences, and participate in online forums to network with peers and industry experts.
2. **Soft Skills:** Develop soft skills such as communication, teamwork, and project management. These are crucial for collaborating effectively in professional settings.
3. **Stay Updated:** Technology evolves rapidly, so it's important to stay informed about the latest trends and advancements in the field.
4. **Certifications:** Obtain relevant certifications to validate your skills and knowledge. Popular certifications include AWS Certified Developer, CompTIA Security+, and Certified Ethical Hacker (CEH).

Career Strategies

1. **Internships and Co-ops:** Gain practical experience through internships and cooperative education programs. These opportunities provide valuable insights into the industry.
2. **Personal Branding:** Create a professional online presence through a personal website, LinkedIn profile, and GitHub repository. Showcase your projects, achievements, and skills.
3. **Mentorship:** Seek out mentors who can provide guidance, support, and advice as you navigate your career path.
4. **Lifelong Learning:** Commit to continuous learning and professional development. The field of computer science is ever-changing, and staying ahead requires a commitment to growth.

Student Workload (SWL)

Structured SWL (h/sem)	75	Structured SWL (h/w)	5
Unstructured SWL (h/sem)	72	Unstructured SWL (h/w)	4.8
Total SWL (h/sem)	150		

Module Evaluation



		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	Quizzes	2	10% (10)	5, 10	LO #1, 2, 10 and 11
	Assignments	2	10% (10)	2, 12	LO # 3, 4, 6 and 7
	Projects / Lab.	1	10% (10)	Continuous	
	Report	1	10% (10)	13	LO # 5, 8 and 10
Summative assessment	Midterm Exam	2 hr	10% (10)	7	LO # 1-7
	Final Exam	2hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

	Material Covered
Week 1	Security and Networking, Security and Networking (Cont.): Network Security Basics. Understanding network threats. Network Troubleshooting
Week 2	Security and Networking (Cont.): Introduction to Network Troubleshooting, Common Network Issues and Symptoms, Network Troubleshooting Tools and Utilities, Security and Networking (Cont.): Using Command-Line Tools for Diagnostics, Identifying and Resolving Connectivity Issues, Diagnosing Network Performance Problems
Week 3	E-Commerce: Concepts of Electronic banking services this includes online banking: ATM and debit card services, E-Commerce (Cont.): Phone banking, SMS banking, electronic alert, Mobile banking.
Week 4	Computer Troubleshooting: Introduction to Computer Troubleshooting, Common Hardware Issues and Solutions, Diagnosing Software Problems, Computer Troubleshooting (Cont.): Hardware , Components: Diagnosis and Repair, Using Safe Mode for Troubleshooting.
Week 5	Computer Troubleshooting (Cont.): Troubleshooting Operating System Issues, Identifying and Resolving Blue Screen Errors, Dealing with Slow Computer Performance, Computer Troubleshooting (Cont.): Virus and Malware Removal Techniques, Updating Drivers and software
Week 6	Introduction to AI: Definition of AI, History of AI, AI Techniques and Approaches, Introduction to AI (Cont.): Key Characteristics of AI, Benefits of AI, Challenges and Ethical considerations, Introduction to AI (Cont.): Challenges and Limitations of AI, The Role of Data in AI Systems
Week 7	introduction to AI (Cont.): AI Tools and Frameworks, The Role of AI in Modern Smartphones: AI-Driven Mobile Technologies, Virtual Assistants (Siri, Google Assistant, Alexa), The Role of AI in Modern Smartphones (Cont.):



	Adaptive Learning, Real-Time Translation Services
Week 8	Midterm Exam
Week 9	The Role of AI in Modern Smartphones (Cont.): The Future of AI in Smartphone Technology, Challenges of Implementing AI in Mobile Devices, Applications and Tools of AI: Overview of AI Applications in Various Industries, Education and Healthcare. Applications and Tools of AI (Cont.): Transportation and Advertising
Week 10	applications and Tools of AI (Cont.): Finance, Robotics and Automation Technologies, Applications and Tools of AI (Cont.): AI in Marketing: Targeting and Personalization
Week 11	Applications and Tools of AI (Cont.): AI in Image and Video Analysis, Smart Cities, Applications and Tools of AI (Cont.): Future Trends in AI Applications and Tools
Week 12	AI and Society: Introduction to AI and Its Societal Impact, The Role of AI in Enhancing Public Safety, AI and Society (Cont.): Cultural Perspectives on AI Adoption, AI and Governance: Policy Implications
Week 13	Ethical Challenges in AI: Introduction to Ethics in AI, Transparency and Explainability of AI Systems, Privacy Concerns in AI Data Usage,
Week 14	Ethical Challenges in AI (Cont.): The Ethical , Implications of Autonomous Systems, Ethics in AI- Driven Marketing and Advertising
Week 15	Ethical Challenges in AI (Cont.): Ethical Considerations in Education, Human Rights and AI Implementation, The Future of AI: Future trends in AI, recent research and emerging technologies
Week 16	Final Exam

Delivery Plan (Weekly Lab. Syllabus)

	Material Covered
Week 1	Introduction to MATLAB environment, interface, and basic commands
Week 2	Lab 2: Variables, data types, and basic mathematical operations
Week 3	Lab 3: Vectors and matrices: creation, manipulation, and indexing
Week 4	Lab 4: Matrix operations and built-in mathematical functions
Week 5	Lab 5: Scripts and basic programming concepts in MATLAB
Week 6	Lab 6: Logical operations and conditional statements (if, else, switch)
Week 7	Lab 7: User-defined functions and function files

Learning and Teaching Resources

	Text	Available in the Library?
Required Texts	Artificial Intelligence: A Modern Approach — Stuart J. Russell & Peter Norvig	
Recommended Texts	Machine Intelligence: Concepts and Mathematics	
Websites	AI Foundations and Applications with MATLAB (Springer)	

APPENDIX:

Group	Grade	Marks (%)	Definition
Success Group (50 - 100)	A - Excellent	90 - 100	Outstanding Performance
	B - Very Good	80 - 89	Above average with some errors
	C - Good	70 - 79	Sound work with notable errors
	D - Satisfactory	60 - 69	Fair but with major shortcomings
	E - Sufficient	50 - 59	Work meets minimum criteria
Fail Group (0 - 49)	FX – Fail	(45-49)	More work required but credit awarded
	F – Fail	(0-44)	Considerable amount of work required
Note:	Note: Marks Decimal places above or below 0.5 will be rounded to the higher or lower full mark (for example a mark of 54.5 will be rounded to 55, whereas a mark of 54.4 will be rounded to 54. The University has a policy NOT to condone "near-pass fails" so the only adjustment to marks awarded by the original marker(s) will be the automatic rounding outlined above.		

**Name of the Instructor:
Bahaa Turki Raji**