MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information معلومات المادة الدراسية					
Module Title	**Computer Programming**			**Module Delivery**	
Module Type	**Core**			☒ **Theory** ☐ **Lecture** ☒ **Lab** ☒ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code	**MPH12009**				
ECTS Credits	**6**				
SWL (hr/sem)	**150**				
Module Level		First	**Semester of Delivery**		2nd
Administering Department		Medical Physics	**College**	Science	
Module Leader	Maysara Aljaf		**e-mail**	Maysara.aljaf@kus.edu.iq	
Module Leader's Acad. Title		Assistant Lecturer	**Module Leader's Qualification**		M.Sc.
Module Tutor	Maysara Aljaf		**e-mail**	Maysara.aljaf@kus.edu.iq	
Peer Reviewer Name		Name	**e-mail**	E-mail	
Scientific Committee Approval Date		21/09/2023	**Version Number**		1.0

Relation with other Modules العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	Computer	**Semester**	First
Co-requisites module	None	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Aims أهداف المادة الدراسية	1. This course provides a manual to operate MATLAB. It presents a detailed course of MATLAB code capabilities required for general programming. 2. MATLAB is a high-performance language of technical computing. It integrates calculation, visualization and programming in an easy-to-use environment where problems and solutions are expressed in writing programs and implementing algorithms through the graphical user interface. MATLAB is an interactive system whose basic data element is an arrays that does not require dimensions. This allows solving many technical computing problems, especially those with matrix and vector formulations. 3. This course includes interactive lectures and practical applications to enable the student to apply algorithms for image processing and apply enhancement methods. In addition, it enables the student to rotate and scale the image by applying many examples.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	The learning outcomes of studying computer programing include: 1. Providing the student with cognitive skills from the basic concepts of programming language and enables them to the skills to run the MATLAB program and dealing with the MATLAB windows and all the types of Statements. 2. Enables students to understood and run all Statements (Loop, Control, Branch), reading and writing data file. 3. Providing the student with cognitive skills to deal with operations with Arrays or Matrices. 4. Providing the student with skills in the technique of dealing with ready-made functions in the language of MATLAB. 5. Providing the student with skills in the Plotting Capabilities, Subplots, 2D Plotting, 3 D Plotting 6. Enables students to deal with the Files: M-files, and functions. 7. They will also learn about algorithms and how to apply them to solve problems. 8. Additionally, they will acquire skills in opening graphical interfaces using the GUIDE UI. 9. Furthermore, the course will cover the types of images and their importance in processing. 10. Students will be provided with digital image processing skills in the MATLAB

	language.
Indicative Contents المحتويات الإرشادية	Indicative content includes the following. <u>Part A</u> Logical Operators, Algorithms and Control Structures, Advantages of algorithm. [5 hrs] <u>Part B</u> MATLAB Windows: Window layout, Command Windows, History Window, Workspace Window, Editor Window, Figure Window, General MATLAB Code: Types of Statements, Rules for Statement Editing, Arithmetic Statement. Constant Value, Variables, Numerical. Variable, Logical Variable, Character Variable, Arrays and Matrices: Index Concept Numerical Arrays and Matrices, Operations on one Arrays or Matrices, N-Dimension Matrices, Logical Arrays, character and String Variables. [12 hrs] <u>Part C</u> Operators, Expression, Loop Statement, Control Statement, Branch Statement, reading and writing data file, Plotting: , Plotting Elementary Function- XY- plotting functions, Generating Sub-Plots, Create Line Plot from Matrix, Specify Line Style, Specify Line Style and Color, Specify Line Width, and Color, Add Title and Axis Labels. [12 hrs] <u>Part D</u> Functions & Files, a) Elementary Mathematical Functions, b) User Defined Functions ,c) Advanced Function Programming ,d) Working with Data Files, Introduction to Graphical User Interfaces (GUI) using GUIDE ,Set the Window Size in GUIDE, Layout the Simple GUIDE UI. [10 hrs] <u>Part E</u> Programming Techniques: Image Types , •Indexed images, •Intensity (or grayscale) images, •RGB (or truecolor) images, Convert signals from an image sensor into digital images, Examples ,Convert Between Image Types, Convert Between Data Types. [8 hrs] <u>Part F</u> Image Rotation and Scale, Examples, Define Image Processing, Image processing in MATLAB, Read Image, Show Image, Examples, Improve clarity, and remove noise and other artifacts, (Enhancement methods in image processing). [8 hrs]

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	When it comes to teaching computer programming, it is important to use strategies that engage students, promote active learning, and facilitate the development of problem-solving and critical-thinking skills. Here are some effective teaching strategies for computer programming: 1. Hands-on coding practice: must provide ample opportunities for students to write code and practice programming through assign coding exercises, projects, and challenges that allow students to apply the concepts they have learned. 2. Pair programming: should encourage students to work in pairs, with one student as the "driver" who writes the code and the other as the "navigator" who reviews the code and offers suggestions. This collaborative approach promotes active learning, communication, and problem-solving skills. 3. Code reviews and feedback: Regularly review and provide feedback on students' code. Offer constructive criticism and guidance on how to improve their programming skills. Encourage students to review and provide feedback to their peers, fostering a culture of collaboration and continuous improvement. 4. Project-based learning: Assign projects that require students to design and develop software applications or solve real-world problems using programming. Project-based learning provides a context for applying programming skills, encourages creativity, and promotes deeper understanding of programming concepts. 5. Demonstrate coding practices: Model good coding practices by writing clean, well-structured code. Explain the reasoning behind coding decisions, such as variable names, code organization, and documentation. Show examples of code refactoring and optimization to help students understand the importance of writing efficient code. 6. Use visual aids and interactive resources: Utilize visual aids, diagrams, flowcharts, and interactive programming environments to explain programming concepts. Visual representations can help students grasp abstract concepts and understand program flow. 7. Real-world examples and applications: Connect programming concepts to real-world examples and applications. Illustrate how programming is used in various domains, such as web development, mobile app development, data analysis, or game development. This helps students understand the practical applications of programming and fosters motivation. 8. Updated with technology trends: Stay informed about the latest programming languages, frameworks, and tools. Integrate current and relevant examples into your teaching to expose students to industry-relevant skills.

Student Workload (SWL) الحمل الدراسي للطالب			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	78	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	5.2
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	72	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	4.8
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	150		

Module Evaluation تقييم المادة الدراسية					
		Time/Number	**Weight (Marks)**	**Week Due**	**Relevant Learning Outcome**
Formative assessment	**Quizzes**	2	5% (5)	4	LO # 1, 2,3,7 and 11
	Assignments	2	10% (10)	5,7,14	LO # 4, 5,6,10,11,12 and 13
	Projects / Lab.	1	20% (20)	Continuous	
	Report	1	5% (5)	10	LO # 8 and 9
Summative assessment	**Midterm Exam**	1hr	10% (10)	9	LO # 1-8
	Final Exam	3hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
	Material Covered
Week 1	Logical Operators, Algorithms and Control Structures, Advantages of algorithm, Examples
Week 2	MATLAB Windows: Window layout, Command Windows, History Window, Workspace Window, Editor Window, Figure Window.
Week 3	General MATLAB Code: Types of Statements, Rules for Statement Editing, Arithmetic Statement. Constant Value, Variables, Numerical. Variable, Logical Variable, Character Variable.
Week 4	Arrays and Matrices: Index Concept Numerical Arrays and Matrices, Operations on one Arrays or Matrices, N-Dimension Matrices, Logical Arrays, character and String Variables.
Week 5	Operators, Expression, Loop Statement, Control Statement, Branch Statement, reading and writing data file.

Week 6	Plotting: , Plotting Elementary Function- XY- plotting functions, Generating Sub-Plots, Create Line Plot from Matrix, Specify Line Style, Specify Line Style and Color, Specify Line Width, and Color, Add Title and Axis Labels.
Week 7	Functions & Files, a) Elementary Mathematical Functions, b) User Defined Functions ,c) Advanced Function Programming ,d) Working with Data Files .
Week 8	Introduction to Graphical User Interfaces (GUI) using GUIDE
Week 9	Set the Window Size in GUIDE, Layout the Simple GUIDE UI
Week 10	Programming Techniques: Image Types , •Indexed images, •Intensity (or grayscale) images, •RGB (or truecolor) images.
Week 11	Convert signals from an image sensor into digital images, Examples
Week 12	Convert Between Image Types, Convert Between Data Types, Examples
Week 13	Image Rotation and Scale, Examples
Week 14	Define Image Processing, Image processing in MATLAB, Read Image, Show Image, Examples
Week 15	Improve clarity, and remove noise and other artifacts, (Enhancement methods in image processing) , Examples
Week 16	**Preparatory week before the final Exam**

Delivery Plan (Weekly Lab. Syllabus) المنهاج الاسبوعي للمختبر	
	Material Covered
Week 1,2	Lab 1: MATLAB Windows, example of Constant Value, Variables, Numerical. Variable, Logical Variable, Character Variable.
Week 3,4	Lab 2: Examples of Arrays and Matrices: Index Concept Numerical Arrays and Matrices, Operations on one Arrays or Matrices, N-Dimension Matrices, Logical Arrays, character and String Variables.
Week 5,6	Lab 3: Examples of Operators, Expression, Loop Statement, Control Statement, Branch Statement, reading and writing data file.
Week 7	Mid Exam
Week 8,9	Lab 4: Examples of Plotting: , Plotting Elementary Function- XY- plotting functions, Generating Sub-Plots, Create Line Plot from Matrix, Specify Line Style, Specify Line Style and Color, Specify Line Width, and Color, Add Title and Axis Labels.
Week10,11	Lab 5: Functions & Files, Introduction to Graphical User Interfaces (GUI) using GUIDE ,Set the Window Size in GUIDE, Layout the Simple GUIDE UI
Week11,12	Lab 6: Programming Techniques: Image Types , •Indexed images, •Intensity (or grayscale) images, •RGB images, Convert signals from an image sensor into digital images, Examples.
Week12,13	Lab 7: Examples of convert between Image Types, Convert Between Data Types, Show Image.

Learning and Teaching Resources
مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts	1- Matlab: Numerical Computing, Tutorials point,2014. 2- Alasdair McAndrew, An Introduction to Digital Image Processing with Matlab, Notes for SCM2511 Image, Processing 1, Semester 1, 2004, School of Computer Science and Mathematics, Victoria University of Technology. 3- The MathWorks, Image Processing Toolbox For Use with MATLAB, Version 2, COPYRIGHT 1993 - 2000.	Yes
Recommended Texts	1- Brian R. Hunt, Ronald L. Lipsman, Jonathan M. Rosenberg, R. Coombes, John E. Osborn, and Garrett J. Stuck, AGuide to MATLAB for Beginners and Experienced Users, Cambridge, University Press,2001	No
Websites		

Grading Scheme
مخطط الدرجات

Group	Grade	التقدير	Marks (%)	Definition
Success Group (50 - 100)	**A -** Excellent	امتياز	90 - 100	Outstanding Performance
	B - Very Good	جيد جدا	80 - 89	Above average with some errors
	C - Good	جيد	70 - 79	Sound work with notable errors
	D - Satisfactory	متوسط	60 - 69	Fair but with major shortcomings
	E - Sufficient	مقبول	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX –** Fail	راسب (قيد المعالجة)	(45-49)	More work required but credit awarded
	F – Fail	راسب	(0-44)	Considerable amount of work required

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.