

DEREE COLLEGE SYLLABUS FOR:											
ITC 3287 Advanced Object Oriented and Functional Programming (Previously ITC 3387) (Updated Fall 2025)			3/1/3 UK LEVEL 5 UK CREDITS: 15								
PREREQUISITES:	ITC 2088 Introduction to Programming ITC 2197 Object Oriented Programming Techniques										
CATALOG DESCRIPTION:	Advanced object-oriented and functional programming concepts; algorithmic and problem-solving techniques; advanced GUI components; event handling; data structures; data persistence; efficiency issues.										
RATIONALE:	The module exposes students to modern software development methodologies and programming principles. Students have the opportunity to design, develop, test and deploy rich client applications, enhancing their understanding of graphical user interface design and their judgement of the effectiveness of different development techniques. An understanding of the basic skills needed in algorithmic design and the interaction between algorithm and data structures in creating efficient code is emphasised.										
LEARNING OUTCOMES:	As a result of taking this course, the student should be able to: 1. Demonstrate understanding of the properties of data structures and select the most appropriate one to solve a computing problem. 2. Explain on a simple problem how functional programming differs from imperative and object-oriented programming. 3. Model software requirements using UML. 4. Design, implement, and test, distributable and maintainable object-oriented GUI applications.										
METHOD OF TEACHING AND LEARNING:	In congruence with the teaching and learning strategy of the college, the following tools are used: • Lectures class discussions and use of generative AI tools to inform course content. • Laboratory sessions, involving training and practice in program design and development. • Homework programming assignments. • Office hours held by the instructor to provide further assistance to students. • Use of the Blackboard site to further support communication, by posting lecture notes, assignment instruction, timely announcements, formative quizzes and online submission of assignments.										
ASSESSMENT:	<table><tr><td colspan="2">Summative:</td></tr><tr><td>1st assessment: Midterm Examination programming problems</td><td>30%</td></tr><tr><td>2nd assessment: Portfolio of student work and oral assessment</td><td>10%</td></tr><tr><td>Final assessment: Programming Project, report</td><td>60%</td></tr></table>			Summative:		1 st assessment: Midterm Examination programming problems	30%	2 nd assessment: Portfolio of student work and oral assessment	10%	Final assessment: Programming Project, report	60%
Summative:											
1 st assessment: Midterm Examination programming problems	30%										
2 nd assessment: Portfolio of student work and oral assessment	10%										
Final assessment: Programming Project, report	60%										

	<p>Formative:</p> <table border="1"> <tr> <td>Continuation/finalization of problems tackled during the in-class hands-on sessions</td><td>0</td></tr> <tr> <td>Midterm formative</td><td>0</td></tr> </table> <p>The formative assessments aim to prepare students for the summative assessments and expose them to teamwork. The 1st summative assessment tests the LOs 1, 2. The 2nd summative assessment tests the LOs 1 - 4. The final summative assessment tests the LOs 1 - 4.</p> <p><i>The final grade for this module will be determined by averaging all summative assessment grades, based on predetermined weights for each assessment. If students pass the final summative assessment, which tests all Learning Outcomes for this module, and the average grade for the module is 40 or above, students are not required to resit any failed assessments.</i></p>	Continuation/finalization of problems tackled during the in-class hands-on sessions	0	Midterm formative	0
Continuation/finalization of problems tackled during the in-class hands-on sessions	0				
Midterm formative	0				
INDICATIVE READING:	<p>REQUIRED READING:</p> <ol style="list-style-type: none"> Charatan Q., Kans A. (2019), <i>Java in Two Semesters</i>, Springer, e-book ISBN: 978-3-319-99420-8. Kendal S., (2019), <i>Object Oriented Programming using C#</i>, Kindle edition (free eBook for students through bookboon) https://bookboon.com/en/object-oriented-programming-using-c-sharp-ebook Instructor's notes. <p>RECOMMENDED READING:</p> <ol style="list-style-type: none"> Deitel P., Deitel H. (2018). <i>Visual C# How to Program</i>, Global Edition, Pearson Education Limited Chin S. et al. (2019). <i>The Definitive Guide to Modern Java Clients with JavaFX</i>, Apress, Kindle edition available Sturm O. (2011). <i>Functional Programming in C#: Classic Programming Techniques for Modern Projects</i>, 1st ed., Wrox Richards M., Ford N. (2020). <i>Fundamentals of Software Architecture: An Engineering Approach</i>, 1st ed., O'Reilly Media, Kindle edition available. 				
INDICATIVE MATERIAL: (e.g. audiovisual, digital material, etc.)	<p>REQUIRED MATERIAL: N/A</p> <p>RECOMMENDED MATERIAL: N/A</p>				
COMMUNICATION REQUIREMENTS:	<p>Daily access to the course's site on the College's Blackboard CMS. Communication using proper written and oral English. Use of word processor and spreadsheet for documentation of assignments.</p>				
SOFTWARE REQUIREMENTS:	<p>MS-Office JetBrains-IntelliJ JetBrains-Clion Microsoft Visual Studio Microsoft Visio</p>				
WWW RESOURCES:	<p>Learning C# https://www.linkedin.com/learning/learning-c-sharp-3?u=80069458</p>				

	<p>Learning JavaFX GUI Development https://www.linkedin.com/learning/learning-javafx-gui-development/provide-a-listview-for-user-selections?u=80069458</p> <p>OOP with JAVA https://classroom.udacity.com/courses/ud283</p>
INDICATIVE CONTENT:	<ol style="list-style-type: none"> 1. Object-Oriented Programming Principles 2. Collections and Data Structures <ol style="list-style-type: none"> a. Linked Lists b. Sets c. Maps d. Stacks, Queues e. Hash Tables 3. Generic and Functional Programming 4. Modelling user requirements with UML 5. Event-Driven Programming and Event Handling 6. Visual Programming and GUI Design 7. Data Persistence 8. Testing methodologies and version control 9. Application packaging and distribution