



University of Tehran
School of Electrical and Computer Engineering

Course:	8101119 – Advanced Programming									
Course type:	EE*						CE*			Credit: 3
	Com	E	P	B	Con	D	SW	HW	IT	
	Required	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Elective	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Level:	Undergraduate <input checked="" type="checkbox"/> Graduate <input type="checkbox"/>									
Co-requisite(s):	None.									
Prerequisite(s):	Introduction to Computer and programming (8101347)									
Prerequisite by topic:	Basic concepts in structural programming and design, being familiar with C language									
Textbook(s):	[1] P.J. Deitel and H.M. Deitel, <i>C++ How to Program</i> . 10 th ed., Prentice-Hall Inc., 2016. [2] P.J. Deitel and H.M. Deitel, <i>Java: How to program</i> . 10 th ed., Prentice Hall Inc., 2016.									
Coordinator:	Ramtin Khosravi, Professor, School of ECE									
Goals:	The goal of this lesson is to provide a variety of ways to produce a quality program. In this regard, after covering the top-down design approach to solve the problem, students will become familiar with the concepts of object-oriented programming as tools for complexity management in medium and large-sized applications. During the course, our focus is on the functionality, test, and debugging of the programs that are implemented via methods such as unit tests and pre-and post-conditions. The emphasis will be on lessons learned than on the structures of a particular programming language. This lesson can be provided in the form of any common object-oriented programming language, such as Java or C++.									
Outcome:	Students who will successfully complete this lesson will be able to <ol style="list-style-type: none"> 1. Solve issues with top-down design techniques, 2. Manage the complexity of the programming problem by defining appropriate classes, 3. Apply appropriate methods such as inheritance and polymorphism, 4. Use the major features of the programming languages' libraries, 5. Use the methods necessary to test and debug the program in order to verify the functionality of the program. 									
Topics:	<ol style="list-style-type: none"> 1) Overview of the basics of programming 2) Top-down design 3) Basic concepts of object-oriented programming: real-world 									

	<p>modeling</p> <ol style="list-style-type: none"> 4) Basic object-oriented programming structures: object, class, method, constructor 5) Inheritance and polymorphism 6) Memory Management - An Introduction to Dynamic Data Structures 7) Generic Programming 8) Handling Errors and Exceptions 9) Input / Output libraries 10) Standard data-structure libraries 11) Create graphical user interface 12) Text processing and threads 13) Introduction to concurrent programming 14) Test and debug the program 								
Computer usage:	The programming environment in an object-oriented language (C++, Java, ...) - Libraries required for items 9 to 13								
Assignments:	About 7 computer assignments								
Projects:	A relatively large project (the last 3 weeks of the semester)								
Grading:	<table style="width: 100%; border: none;"> <tr> <td style="width: 60%;">Assignments:</td> <td style="text-align: right;">20%</td> </tr> <tr> <td>Projects:</td> <td style="text-align: right;">10%</td> </tr> <tr> <td>Midterm exams:</td> <td style="text-align: right;">35%</td> </tr> <tr> <td>Final exam:</td> <td style="text-align: right;">35%</td> </tr> </table> <p>*Students are required to complete at least half of the written tests' grades.</p>	Assignments:	20%	Projects:	10%	Midterm exams:	35%	Final exam:	35%
Assignments:	20%								
Projects:	10%								
Midterm exams:	35%								
Final exam:	35%								
Further readings:	<p>[1] B. Stroustrup, <i>Programming Principles and Practice Using C++</i>. 2nd ed., Addison-Wesley, 2014.</p> <p>[2] B. Eckel, <i>Thinking in Java</i>. 4th Edition, Prentice Hall, 2006.</p>								
Prepared by:									
Date:									

*EE: Electrical Engineering		CE: Computer Engineering	
Com	Communications	SW	Software
E	Electronics	HW	Hardware
P	Power	IT	Information Technology
B	Bioelectronics		
Con	Control		
D	Digital System		