CENG 110 - PROGRAMMING AND COMPUTING II

Course Code	Course Name	Semester	
CENG 110	Programming And Computing	Fall ☐ Spring ☒ Summer ☐	
Hours		**Credit**	**ECTS**
Theory	**Practice** / **Lab**	3	6
3	0 / 0		

Course Details	
Department	Software Engineering
Course Language	English
Course Level	Undergraduate ☒ Graduate ☐
Mode of Delivery	Face to Face ☒ Online ☐ Hybrid ☐
Course Type	Compulsory ☒ Elective ☐
Course Objectives	By the end of this course, students will be able to: 1. Develop a strong understanding of C and C++ programming concepts, including procedural and object-oriented paradigms. 2. Implement efficient memory management and data structures, such as linked lists and self-referential structures. 3. Apply secure programming practices and understand best practices for writing safe and optimized code. 4. Utilize file handling techniques, including sequential and random-access file processing. 5. Work with multi-file compilation, input/output redirection, and preprocessor directives to create modular programs. 6. Design and develop scalable applications using functions, templates, and the C++ Standard Library.
Course Content	This course provides a structured introduction to C and C++ programming, covering both fundamental and advanced topics. Students will begin with character and string manipulation, formatted input/output, data structures, dynamic memory allocation, and file handling in C. They will also explore bitwise operations, preprocessor directives, and secure programming techniques to write efficient and maintainable code. In the second part, the course transitions to C++ programming, introducing functions, references, templates, and object-oriented programming (OOP) concepts such as encapsulation, inheritance, and polymorphism. The C++ Standard Library will be utilized to develop scalable and modular applications. Hands-on exercises, coding projects, and assessments will reinforce key concepts and practical skills needed for real-world software development.
Course Method/ Techniques	Lecture ☒ Question & Answer ☒ Presentation ☒ Discussion ☒

Prerequisites/ Corequisites	
Work Placement(s)	No work placement is required for this course

Textbook/References/Materials

Textbook :

1. C: How to Program with Introduction to C++, Global Edition, Eighth Edition, H. M. Deitel and P. J. Deitel, Pearson

References :

1. Cisco Networking Academy - C++ Essentials 1 (Build your C++ programming skills and prepare for CPE – C++ Certified Entry-Level Programmer certification.)
2. Cisco Networking Academy - C++ Essentials 2 (Take your C++ skills to the next level and prepare for CPA – C++ Certified Associate Programmer certification.)

Course Category				
Mathematics and Basic Sciences	☐		Education	☐
Engineering	☒		Science	☐
Engineering Design	☐		Health	☐
Social Sciences	☐		Profession	☐

Weekly Schedule

No	Topics	Materials/Notes
1	C Characters and Strings	Deitel Chapter 8.1-8.5
2	C String Manipulation	Deitel Chapter 8.6-8.11
3	Formatted Input/Output in C	Deitel Chapter 9.1-9.9
4	C Structures and Unions	Deitel Chapter 10.1-10.8
5	Bitwise Operations and Enumerations in C	Deitel Chapter 10.1-10.13
6	C File Processing (Sequential Access)	Deitel Chapter 11.1-11.4
7	Random-Access Files and Secure Programming in C	Deitel Chapter 11.1-11.4
8	Midterm Exam	
9	Data Structures: Self-Referential Structures, Dynamic Memory Allocation, and Linked Lists in C	Deitel Chapter 12.1-12.4
10	Advanced C Preprocessor Directives and Secure Programming	Deitel Chapter 13
11	Input/Output Redirection and Multi-File Compilation	Deitel Chapter 14.1-14.5
12	Program Control and Dynamic Memory Allocation	Deitel Chapter 14.6-14.10

13	Fundamentals of C++ and Basic Programming Concepts	Deitel Chapter 15.1-15.6
14	Functions, References, and Templates in C++	Deitel Chapter 15.7-15.13
15	Object-Oriented Programming and the C++ Standard Library	Deitel Chapter 15.14-15.16
16	Final Exam	

Assessment Methods and Criteria		
In-term studies	**Quantity**	**Percentage**
Attendance		
Lab		
Practice		
Fieldwork		
Course-specific internship		
Quiz/Studio/Criticize		
Homework		20%
Presentation / Seminar		
Project		
Report		
Seminar		
Midterm Exam		30%
Final Exam		50%
	Total	**100%**
Contribution of Midterm Studies to Success Grade		**50%**
Contribution of End of Semester Studies to Success Grade		**50%**
	Total	**100%**

ECTS Allocated Based on Student Workload			
Activities	**Quantity**	**Duration (Hrs)**	**Total Workload**
Course Hours	14	3	42
Lab			
Practice			
Fieldwork			
Course-specific Work Placement			
Out-of-class study time	14	3	42
Quiz/Studio/Criticize			
Homework	2	10	20
Presentation / Seminar			
Project			
Report			
Midterm Exam and Preparation for Midterm	1	20	20
Final Exam and Preparation for Final Exam	1	20	20

Total Workload	**130**
Total Workload / 25	**5,76**
ECTS Credit	**6**

Course Learning Outcomes	
No	**Outcome**
L1	Understand the core principles of C++ programming, including syntax, semantics, and object-oriented concepts like encapsulation, inheritance, and polymorphism.
L2	Analyze and implement advanced features such as operator overloading, exception handling, and dynamic memory management for robust and efficient applications.
L3	Design modular and reusable programs using classes, objects, templates, and inheritance hierarchies to address real-world problems.
L4	Utilize C++ Standard Library components such as vector, streams, and file handling to effectively manage data and perform input/output operations.
L5	Develop object-oriented systems by applying design principles and leveraging Unified Modeling Language (UML) for system visualization and architecture.

Contribution of Course Learning Outcomes to Program Competencies/Outcomes

Contribution Level: 1: Very Slight, 2: Slight, 3: Moderate, 4: Significant, 5: Very Significant

	P1	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**	**P9**	**P10**	**P11**	**P12**	**P13**	**P14**	**P15**	**Total**
L1	5	5	-	4	-	-	-	-	-	-	-	-	-	-	-	14
L2	5	5	4	5	-	-	-	-	-	-	-	-	-	-	-	19
L3	5	5	5	5	-	-	-	-	-	-	-	-	-	-	-	20
L4	5	5	5	5	4	-	-	-	-	-	-	-	-	-	-	24
L5	4	5	5	5	-	4	4	-	-	-	-	-	-	-	-	27
Total																104