Identification	**Subject**	***CMS 105:*** Fundamentals of Computer Programming - 6ECTS
	Department	Computer Science
	Program	Undergraduate
	Term	Spring, 2023
	Instructor	Azeem
	E-mail:	hm.azeem.akram@outlook.com
	Classroom/hours	**402N/** 1st Session: 8:30-10:00 ‖ 2nd Session: 10:10-11:40, **Day**: Wednesday
Prerequisites	English proficiency	
Language	English	
Compulsory/Elective	Required	
Required textbooks and course materials	***Core textbooks:*** 1. "C++ How to Program" by Deitel & Deitel 2. "C Programming Language 2nd Edition" by Brian W. Kernighan and Dennis M. Ritchie 3. "C Programming Absolute Beginner's Guide 3rd Edition" by Greg Perry and Dean Miller	
Course outline	1. Introduction to Programming: Overview of the importance of programming languages and the role of C in creating software. 2. Data Types and Variables: Overview of basic data types in C, including integers, floats, and characters, and how to declare and use variables. 3. Control Structures: Introduction to control structures, such as if/else statements, loops, and switch statements, and how to use them to control the flow of a program. 4. Functions: Overview of functions, including function prototypes, definitions, and calls, and how to use functions to break a program into smaller, manageable parts. 5. Arrays and Structures: Introduction to arrays and structures in C and how to manipulate and process data using these structures. 6. Pointers: Introduction to pointers, including pointer arithmetic and the use of pointers to manipulate memory locations. 7. File Input/Output: Overview of file input/output operations in C, including how to read and write data to files. 8. Dynamic Memory Allocation 9. Preprocessor Directives 10. Advanced Topics	
Course objectives	1. To appreciate the need for a programming language 2. To introduce the concept and usability of the structured programming methodology 3. To develop proficiency in making useful software using the C language	

Learning outcomes	The learning outcomes are as follows: 1. Understanding of the basics of programming and computer science: You will gain an understanding of how programs are written and executed, as well as the fundamental concepts of algorithms and data structures. 2. Proficiency in C programming: You will learn how to write and run C programs, use variables and data types, control program flow with conditionals and loops, work with functions and pointers, and more. 3. Familiarity with the standard library: You will learn about the C standard library and its functions, including those for input/output, string manipulation, and memory allocation. 4. Knowledge of best practices and debugging techniques: You will learn how to write efficient and well-organized code, as well as how to debug programs and find and resolve errors. 5. Preparation for further study: A solid foundation in C programming will provide a strong foundation for further study in other programming languages, as well as more advanced topics in computer science such as algorithms and data structures. 6. Hands-on experience: You will gain practical experience by writing and debugging programs, as well as solving programming problems and exercises. This will help you build your confidence and solidify your understanding of the concepts.

Teaching methods	**Lecture**		x
	Group discussion		x
	Experiential exercise		x
	Lab		x
	Case analysis		
	Course paper		x
	Others		
Evaluation	**Methods**	**Date/deadlines**	**Percentage (%)**
	Midterm Exam		30
	Case studies		
	Class Participation		10
	Assignment and quizzes		(20+10=30)
	project		
	Presentation/Group Discussion		
	Final Exam		30
	Others		
	Total		100

Policy	**Preparation for class** The lecture material will focus on the major points introduced in the text. Reading the assigned chapters and having some familiarity with them before class will greatly assist your understanding of the lecture. After the lecture, you should study your notes and work relevant problems. **Quizzes:** There will be three quizzes in class, and the highest score of two will be considered. **Group coding assignments**: Grades will be assigned based on the following five principles: 1. Readability: Code should be easy to read and understand, with clear variable and function names and proper indentation and comments. 2. Maintainability: Code should be well structured and modular, making it easy to modify or extend in the future. 3. Performance: Code should be optimized for speed and efficiency, without sacrificing readability and maintainability. 4. Correctness: Code should be free of bugs and errors and should handle all possible inputs and edge cases correctly. 5. Viva presentation.

- **Withdrawal (pass/fail)**
  This course strictly follows grading policy of the School of Engineering and Applied Science. Thus, a student is normally expected to achieve a mark of at least 60% to pass. In case of failure, he/she will be required to repeat the course the following term or year.

- **Cheating/plagiarism**
  Cheating or other plagiarism during the Quizzes, Mid-term and Final Examinations will lead to paper cancellation. In this case, the student will automatically get zero (0), without any considerations.

- **Professional behavior guidelines**
  The students shall behave in the way to create favorable academic and professional environment during the class hours. Unauthorized discussions and unethical behavior are strictly prohibited.

- **Ethics**
  Students should not arrive in late to class.
  All cell phones must be turned off and stowed away before entering class.
  Use of any electronic devices is not allowed in the classroom and violators will be punished accordingly.

WK	Date/Day (tentative)	Topics	Textbook/Assignments
1		**Introduction and Basics** • Introduction to programming and C • Basic data types and variables • Review and practice problems	Lecture Slides & Classroom Practice Readings:1.1-1.6, 1.11-1.13
2		• Operators and expressions • Input and Output functions • Review and practice problems	Lecture Slides & Classroom Practice Readings:1.19-1.22
3		**Control Structures** • If-Else statements • Switch statements • Review and practice problems	Lecture Slides & Classroom Practice Readings:2.16, 2.4-2.6, 2.19-2.20 **Assignment#01 Released**
4		• While and Do-While loops • For loops • Review and practice problems	**Quiz#01** Lecture Slides & Classroom Practice Readings:2.7-2.9,2.11,2.12,2.14,2.15,2.17,2.19.
5		**Functions** • Defining and calling functions • Pass by value and pass by reference • Review and practice problems	Lecture Slides & Classroom Practice Readings:3.1-3.6
6		• Function prototypes and header files • Recursion • Review and practice problems	**Assignment#01 Viva** Lecture Slides & Classroom Practice Readings:3.7,3.11, 3.12,3.14,3.17
7		**Arrays and Strings** • Arrays • Multidimensional arrays • Strings • String manipulation functions • Review and practice problems	**Quiz#02** Lecture Slides & Classroom Practice Readings:4.2-4.9, 5.29-5.34

8		**Midterm Exam**	
9		**Pointers** • Introduction to pointers • Pointer arithmetic • Review and practice problems	Lecture Slides & Classroom Practice Readings:5.1-5.6
10		• Pointers and arrays • Pointers and functions • Review and practice problems	**Assignment#02 Released** Lecture Slides & Classroom Practice Readings:5.7-5.10
11		**Structures and Unions** • Structures • Initializing structures • Accessing structure members • Nested structures and unions • Review and practice problems	Lecture Slides & Classroom Practice Readings:6.2-6.4, 16.3-16.5
12		**File Input/Output** • Opening and closing files • Reading and writing data • Binary file I/O • Command line arguments • Review and practice problems	**Quiz#03** Lecture Slides & Classroom Practice Readings:14.3-14.6
13		**Dynamic Memory Allocation** • Malloc, Calloc, and Realloc • Free function • Dynamic arrays • Dynamic structures	Lecture Slides & Classroom Practice Readings: Chapter 7.6-7.8
14		**Preprocessor Directives** • Macros and #define • File inclusion with #include • Conditional Compilation with #ifdef • Error handling with #error and #pragma • Review and practice problems	**Assignment#02 Viva** Lecture Slides & Classroom Practice Readings: Chapter 17
15		**Advanced Topics** • Bitwise operators and bit manipulation • Enumerated data type • Type casting • Standard libraries (math.h, string.h, stdio.h) • Review and practice problems	Lecture Slides & Classroom Practice Readings: Chapter 16
		Final Exam	

Note: All the readings mentioned above are from the book "C++ How to Program" by Deitel & Deitel.