

## **C++ Programming - Beginner to Pro**

### **Course Policies & Syllabi, 2024**

Course Name	C++ Programming - Beginner to Pro
Duration of the Course in Weeks	26 weeks
Duration of the Course in Academic Hours	156 (of which 60% practice hours)
Tuition Fee	600,000 Kazakhstani Tenge (100,000 KZT per month)
Learning Format	Online – pre-recorded lectures + live practice hours (Google Meet)
Learning Platform	In-house LMS + Google Meet
Frequency of Classes	Weekly lectures + bi-weekly practice (combined 6 hours)
Skills / Results Upon Completion of Training	Proficiency in C++ programming language
Qualification Upon Completion of Training	Certificate of Completion in "C++ Programming - Beginner to Pro"
Methodological Basis of the Course	Emphasis on hands-on practice through coding exercises and projects.
The Ratio of Theory to Practice	60-40 (60% practice, 40% theory)
Teaching Staff	Sufyan Mustafa bin Uzayr
Tracking the Development of Students	Assessment and Evaluation Feedback Mechanisms Coding Portfolios Peer and Self-Assessment
Student Performance Evaluation Methods	Midterms + Coding Tests + Final Exam

## Course Description

This comprehensive course is designed to take learners from beginner to advanced levels in C++ programming. Starting with the basics of syntax and control structures, participants will progress through topics such as object-oriented programming principles, memory management, and advanced features like templates and exception handling. Through practical exercises and projects, students will gain hands-on experience, enabling them to confidently tackle a variety of programming challenges and applications upon completion. Whether aspiring to develop software, games, or system-level applications, this course equips individuals with the necessary skills to excel in the world of C++ programming.

## Course Duration

Number of weeks	26
Number of academic hours	156 (of which 60% practice hours)

## Pre-requisites

- As the course materials, lectures, and instructions are primarily delivered in English, a strong understanding of the English language is essential for effective comprehension and communication.
- Fundamental logical and reasoning abilities are required to understand programming concepts and solve problems efficiently.

## Skills Taught

- Proficiency in C++ syntax, data types, and control structures
- Mastery of Object-Oriented Programming (OOP) principles
- Advanced features including templates, exception handling, and file I/O

- Problem-solving and algorithmic thinking skills
- Hands-on experience through coding exercises and projects
- Development of software applications ranging from desktop apps to games
- Code optimization techniques for performance and efficiency
- Collaboration and communication skills for working in teams and contributing effectively

## Learning Plan

Week	Topic	Notes
1	Introduction to C++ Programming. Overview of C++ language and its importance in software development. Brief history and evolution of C++ Introduction to Integrated Development Environments (IDEs) for C++ development.	
2	Basic Syntax and Structure. Understanding basic syntax rules such as comments, statements, and identifiers. Overview of data types including integers, floating-point numbers, characters, and boolean. Declaration and initialization of variables in C++.	
3	Data Types and Variables. Exploring fundamental data types like int, float, double, char, and bool. Understanding variable scope and lifetime. Declaring and initializing variables of different types in C++.	
4	Operators and Expressions. Overview of arithmetic, relational, logical, bitwise, and assignment operators. Expression evaluation and operator precedence. Using operators effectively in C++ expressions.	

5	Control Structures: If-Else, Switch. Understanding conditional statements with if, else, and else if Implementing switch-case statements for multiple branching. Writing conditional expressions to control program flow.	
6	Loops: For, While, Do-While. Implementing iterative loops with for, while, and do-while constructs. Understanding loop termination conditions and loop control statements. Using loops for repetitive tasks and iteration over data structures.	CT-1
7	Functions and Modular Programming. Introduction to functions and their role in modular programming. Defining and invoking functions with different return types and parameters. Understanding function prototypes and function overloading.	
8	Arrays and Strings. Declaring and initializing arrays of different types. Accessing array elements and performing array operations. Introduction to strings as character arrays and string manipulation functions in C++.	
9	Pointers and Dynamic Memory Allocation. Understanding pointers and memory addresses. Dynamic memory allocation with new and delete operators. Pointer arithmetic and memory management techniques.	
10	Introduction to Object-Oriented Programming (OOP). Overview of OOP concepts: classes, objects, inheritance, polymorphism, and encapsulation. Understanding the benefits of OOP paradigm in software development. Comparison of procedural programming with OOP principles.	

<b>11</b>	Classes and Objects. Defining classes and creating objects in C++. Encapsulating data and methods within a class. Access control modifiers: public, private, and protected.	
<b>12</b>	Constructors and Destructors. Understanding constructors and their role in object initialization. Defining parameterized and default constructors. Implementing destructors for resource cleanup and memory management.	
<b>13</b>	Midterm-1	Midterm-1
<b>14</b>	Inheritance and Polymorphism. Implementing inheritance to create hierarchical relationships between classes. Understanding base and derived classes. Polymorphism through function overriding and virtual functions.	
<b>15</b>	Encapsulation and Abstraction. Encapsulation: bundling data and methods within a class Access control modifiers: public, private, and protected. Implementing encapsulation for data hiding and abstraction.	
<b>16</b>	Function Overloading and Operator Overloading Defining multiple functions with the same name but different parameters. Overloading operators to work with user-defined types. Implementing function and operator overloading for code reusability.	
<b>17</b>	Templates and Generic Programming. Introduction to templates for generic programming. Creating function templates and class templates. Benefits of template programming for code flexibility and performance.	
<b>18</b>	Exception Handling. Overview of exception	CT-2

	handling mechanism in C++. Handling runtime errors using try-catch blocks. Throwing and catching exceptions for robust error handling	
<b>19</b>	File Input and Output (I/O). Reading from and writing to files in C++. Using file streams for file I/O operations. File handling techniques for data persistence and manipulation.	
<b>20</b>	Standard Template Library (STL). Overview of STL containers: vectors, lists, maps, sets, etc. Algorithms provided by STL for common data manipulation tasks. Using iterators to traverse and manipulate STL containers.	
<b>21</b>	Advanced Topics in Memory Management. Memory management techniques: stack vs. heap memory. Dynamic memory allocation using new and delete operators. Memory leaks and memory corruption prevention strategies.	
<b>22</b>	Smart Pointers. Introduction to smart pointers: unique_ptr, shared_ptr, and weak_ptr. Automatic memory management and resource cleanup with smart pointers. Benefits of smart pointers over raw pointers in C++ programming.	
<b>23</b>	Multithreading and Concurrency. Introduction to multithreading and concurrency concepts reating and managing threads in C++ using the <thread> library. Synchronization mechanisms: mutexes, condition variables, and locks.	
<b>24</b>	Debugging and Testing Techniques. Techniques for debugging C++ programs: print debugging, breakpoints, and tracing. Unit testing frameworks for C++: Google Test, Catch, etc. Writing effective test cases for C++ code coverage and quality	

	assurance.	
<b>25</b>	End Term (Midterm-2)	Midterm-2
<b>26</b>	Final Project: OpenGL and High-Performance Graphics with C++	
<b>Finals Period</b>		Final Exam

## Student Evaluation Criteria

Component	Description	When	Points
Class Test	(a) Computer-based (b) Online	6th and 18th week	each worth 15 points → 30 points
Midterms	(a) Computer-based (b) Online	13th and 25th week	each worth 15 points → 30 points
Final Exam	(a) Computer-based (b) Online	Finals Period	40 points
Total			100 points

Letter Grade	Points	Grade Remarks
A	95-100	<b>Excellent</b>
A-	90-94	
B+	85-89	<b>Good</b>
B	80-84	

B-	75-79	
C+	70-74	
C	65-69	<b>Satisfactory</b>
C-	60-64	
D+	55-59	
D	50-54	
FX	25-49	<b>Fail</b>
F	0-24	

## Course Fee

600,000 Kazakhstani Tenge (100,000 KZT per month)

## Teaching Staff

Sufyan Mustafa bin Uzayr
Telegram Username - @sufyanbinuzayr
Site: <a href="https://sufyanism.com/">https://sufyanism.com/</a>
Google Scholar Profile - <a href="https://scholar.google.com/citations?user=CzF2dJIAAAAJ">https://scholar.google.com/citations?user=CzF2dJIAAAAJ</a>
Sufyan Mustafa bin Uzayr is a skilled teacher with extensive experience in the IT industry, having taught for over 5 years. He is proficient in various software tools and programming paradigms. Having authored 60+ books, their expertise in Sufyan has taught and worked at universities and colleges in 3 different continents.



## Student Selection Criteria

To be admitted to the program, students are required to pass an Aptitude Test, consisting of 25 questions. The Test shall be conducted online, the dates of which shall be communicated to the students via email.

Additionally, the Test topics shall include General Reasoning and Logic, Mathematical Skills (School Grade-8 level).

It is a pre-requisite for students to possess general verbal and oral English skills.

## Training Format

Online – pre-recorded lectures + live practice hours (Google Meet)

## Use of Technology in Teaching

Zeba Academy uses its own flagship Learning Management System, that comes with features such as Pre-recorded Video Lessons, Advanced Quizzes, Flashcards, Image Hotspots Groups and Cohorts, AI-based adaptive learning algorithm, and guided practice materials in the form of ebooks and other content.

## Contact

Zeba Academy Help Desk for Students

Telegram and WhatsApp: **+7 705 435 2915**

