A Project Report

on

Quiz Game CLI C++



by

Muhammad Bin Khalid BCPE243046

A Project Report submitted to the

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

in partial fulfillment of the requirements for the degree of

BACHELORS OF SCIENCE IN ELECTRICAL ENGINEERING

Faculty of Engineering

Capital University of Science & Technology,

Islamabad

June, 2025

Copyright © 2025 by CUST Student

All rights reserved. Reproduction in whole or in part in any form requires the prior written permission of Muhammad Bin Khalid or designated representative.

# DECLARATION

It is declared that this is an original piece of our own work, except where otherwise acknowledged in text and references. This work has not been submitted in any form for another degree or diploma at any university or other institution for tertiary education and shall not be submitted by us in future for obtaining any degree from this or any other University or Institution.

Muhammad Bin Khalid

Reg. No. BCPE243046

June, 2025

# CERTIFICATE OF APPROVAL

It is certified that the project titled “Quiz Game ClI C++” carried out by Muhammad Bin Khalid, Reg. No. BCPE243046, under the supervision of Mr. Muneeb Ahmed, Capital University of Science & Technology, Islamabad, is fully adequate, in scope and in quality, as a final year project for the degree of BS Electrical Engineering.

Supervisor: --------------------------------------

Mr. Muneeb Ahmed

Lecturer

Department of Electrical and Computer Engineering

Faculty of Engineering

Capital University of Science & Technology, Islamabad

HoD: --------------------------------------

Dr. Noor Mohammad Khan

Professor

Department of Electrical and Computr Engineering

Faculty of Engineering

Capital University of Science & Technology, Islamabad

# ACKNOWLEDGMENT

We would like to thank “Mr. Muneeb Ahmed” whose guidance helped to complete our report in the given time. Last but not the least, this project cannot be completed without the effort and co-operation of group members.

# ABSTRACT

This project presents the development of a C++-based command-line quiz game designed as an educational tool to reinforce fundamental programming concepts. The application implements a dynamic quiz system featuring a question bank with multiple-choice questions, random question shuffling, user input validation, a scoring mechanism, and persistent user history tracking. Built using object-oriented programming principles, the QuizManager class encapsulates core functionalities, including file handling for storing user performance data and Windows API for enhanced console interaction with color-coded output. The system employs the Fisher-Yates shuffle algorithm for unbiased question randomization and a timer to track performance metrics. Developed using tools like GCC/MSVC compilers and Visual Studio Code, the project demonstrates the practical application of C++ constructs such as classes, arrays, and file operations. It serves as an effective learning resource for students and beginners, offering insights into structured programming and software development practices. The quiz game is limited to Windows platforms due to API dependencies but provides a robust, user-friendly experience for educational assessment.**TABLE OF CONTENTS**

[DECLARATION iii](#_Toc200878523)

[CERTIFICATE OF APPROVAL iv](#_Toc200878524)

[ACKNOWLEDGMENT v](#_Toc200878525)

[ABSTRACT vi](#_Toc200878526)

[LIST OF ACRONYMS/ABBREVIATIONS ix](#_Toc200878527)

[INTRODUCTION 0](#_Toc200878528)

[1.1 Overview 0](#_Toc200878529)

[1.2 Project Idea 0](#_Toc200878530)

[1.3 Purpose of the Project 0](#_Toc200878531)

[1.4 Project Specifications 0](#_Toc200878532)

[1.5 Applications of the Project 0](#_Toc200878533)

[1.5.1 Educational Assessment Tool 0](#_Toc200878534)

[1.6 Project Plan 1](#_Toc200878535)

[1.6.1 Project Milestones 1](#_Toc200878536)

[1.7 Report Organization 1](#_Toc200878537)

[LITERATURE REVIEW 2](#_Toc200878538)

[2.1 Background Theory 2](#_Toc200878539)

[2.2 Command-Line Applications 2](#_Toc200878540)

[2.3 Quiz Systems 2](#_Toc200878541)

[2.3.1 Classification of Quiz Systems 2](#_Toc200878542)

[2.4 Related Technologies 2](#_Toc200878543)

[2.4.1 C++ Programming Language 2](#_Toc200878544)

[2.4.2 Object-Oriented Programming (OOP) 2](#_Toc200878545)

[2.4.3 File Handling in C++ 2](#_Toc200878546)

[2.4.4 Windows API for Console Manipulation 3](#_Toc200878547)

[2.4.5 Randomization Algorithms 3](#_Toc200878548)

[2.5 Limitations and Bottlenecks of the Existing Work 3](#_Toc200878549)

[2.6 Problem Statement 3](#_Toc200878550)

[2.7 Summary 3](#_Toc200878551)

[PROJECT DESIGN AND IMPLEMENTATION 4](#_Toc200878552)

[3.1 Proposed Design Methodology 4](#_Toc200878553)

[3.1.1 Class Structure: QuizManager 4](#_Toc200878554)

[3.2 Analysis Procedure 4](#_Toc200878555)

[3.2.1 User Input Validation 4](#_Toc200878556)

[3.3 Design of the Project Software/Algorithm 4](#_Toc200878557)

[3.3.1 Question Shuffling Logic 4](#_Toc200878558)

[3.3.2 Scoring System 4](#_Toc200878559)

[3.4 Implementation of Features 4](#_Toc200878560)

[3.4.1 Timer and Performance Metrics 4](#_Toc200878561)

[3.4.2 User History Tracking 5](#_Toc200878562)

[3.5 Summary 5](#_Toc200878563)

[TOOLS AND TECHNIQUES 6](#_Toc200878564)

[4.1 Standard PC/Laptop 6](#_Toc200878565)

[4.2 Tools 6](#_Toc200878566)

[4.2.1 C++ Compiler (GCC/MSVC) 6](#_Toc200878567)

[4.2.2 Visual Studio Code 6](#_Toc200878568)

[4.2.3 Windows API (for Console Features) 6](#_Toc200878569)

[4.3 Summary 6](#_Toc200878570)

[Source Code Snippets 7](#_Toc200878571)

[Sample User History File 8](#_Toc200878572)

# LIST OF ACRONYMS/ABBREVIATIONS

CLI Command Line Interface

STL Standard Template Library

API Application Programming Interface

GUI Graphical User Interface

OOP Object-Oriented Programming

CSV Comma-Separated Values

IDE Integrated Development Environment

Chapter 1

# INTRODUCTION

## 1.1 Overview

This report details the development of a C++-based command-line quiz game designed to demonstrate fundamental programming concepts. The application serves as an educational tool, allowing users to test their knowledge through an interactive quiz system. It incorporates features such as question shuffling, user history tracking, and performance metrics, all implemented using core C++ principles and Windows API for console manipulation.

## 1.2 Project Idea

The project idea originated from the need to create an engaging educational tool that reinforces C++ programming concepts. The quiz game provides a practical application of data structures, file handling, and object-oriented programming, making it an effective learning resource for students and beginners in C++ programming.

## 1.3 Purpose of the Project

The primary purpose is to demonstrate the application of C++ programming constructs in developing a fully functional quiz game. It aims to provide a hands-on learning experience, showcasing how classes, arrays, functions, and file operations can be integrated to create an interactive application.

## 1.4 Project Specifications

The quiz game is designed with the following specifications:

* Platform: Windows-based command-line interface • Language: C++ with standard libraries and Windows API
* Features:
  + Question bank with multiple-choice questions
  + Random question shuffling
  + User input validation
  + Timer for performance tracking
  + User history storage in text files
  + Color-coded console output
* Scoring: 3 points per correct answer, with a total of 5 questions

## 1.5 Applications of the Project

### 1.5.1 Educational Assessment Tool

The quiz game serves as an effective tool for educational institutions to assess students’ understanding of C++ programming concepts. It can be used in classroom settings or self-study environments to reinforce learning through interactive testing.

## 1.6 Project Plan

### 1.6.1 Project Milestones

The project was developed in the following phases:

1. Requirement Analysis (Week 1): Defined project scope and specifications.
2. Design Phase (Weeks 2-3): Developed class structure and algorithm design.
3. Implementation (Weeks 4-6): Coded core functionalities and integrated features.
4. Testing and Debugging (Week 7): Conducted thorough testing to ensure reliability.
5. Documentation (Week 8): Compiled this report and finalized deliverables.

## 1.7 Report Organization

This report is structured as follows:

* Chapter 1: Introduction to the project and its objectives.
* Chapter 2: Literature review of related technologies and systems.
* Chapter 3: Detailed design and implementation of the quiz game.
* Chapter 4: Tools and techniques used in development.
* References: Sources cited in the report.
* Appendices: Source code snippets and sample user history file.

Chapter 2

# LITERATURE REVIEW

## 2.1 Background Theory

Quiz systems have been widely used in educational settings to evaluate knowledge retention. Command-line applications, while simple, provide a robust platform for implementing such systems due to their minimal resource requirements and direct user interaction.

## 2.2 Command-Line Applications

Command-line applications are text-based interfaces that allow users to interact with software through typed commands. They are lightweight, efficient, and suitable for educational tools like quiz systems, as demonstrated by similar projects in C++ [1].

## 2.3 Quiz Systems

Quiz systems range from simple text-based applications to complex web-based platforms. They typically include features like question randomization, scoring, and user feedback, which are critical for effective learning [2].

### 2.3.1 Classification of Quiz Systems

Quiz systems can be classified as:

* Static: Fixed question sets with no randomization.
* Dynamic: Questions shuffled or selected randomly.
* Adaptive: Questions adjust based on user performance.

This project implements a dynamic quiz system with randomized question selection.

## 2.4 Related Technologies

### 2.4.1 C++ Programming Language

C++ is a versatile, high-performance language ideal for system-level programming. Its support for object-oriented programming and standard libraries makes it suitable for developing quiz applications [3].

### 2.4.2 Object-Oriented Programming (OOP)

OOP principles such as encapsulation, modularity, and reusability were employed to design the QuizManager class, ensuring maintainable and scalable code [4].

### 2.4.3 File Handling in C++

File handling in C++ enables persistent storage of user data, such as quiz results and history, using streams like fstream [5].

### 2.4.4 Windows API for Console Manipulation

The Windows API provides functions like SetConsoleTextAttribute for enhancing console output with colors, improving user experience [6].

### 2.4.5 Randomization Algorithms

Randomization algorithms, such as the Fisher-Yates shuffle, ensure unbiased question selection, enhancing the quiz’s fairness [7].

## 2.5 Limitations and Bottlenecks of the Existing Work

Existing quiz systems often lack:

* Robust user input validation
* Persistent user history storage
* Cross-platform compatibility

This project addresses these by implementing input validation and file-based history storage but is limited to Windows due to API usage.

## 2.6 Problem Statement

The challenge was to develop a C++-based quiz game that is user-friendly, robust, and educational, addressing limitations like lack of randomization and persistent storage in existing systems.

## 2.7 Summary

The literature review highlights the relevance of command-line quiz systems, the suitability of C++ and OOP, and the importance of randomization and file handling for a robust application.

Chapter 3

# PROJECT DESIGN AND IMPLEMENTATION

## 3.1 Proposed Design Methodology

### 3.1.1 Class Structure: QuizManager

The core of the project is the QuizManager class, which encapsulates all quiz functionalities:

* Data Members: Arrays for questions, options, and correct answers; variables for score, user name, and timer.
* Methods: Functions for loading questions, shuffling, processing user input, and storing results.

## 3.2 Analysis Procedure

The system was analyzed to ensure robust input handling, efficient question randomization, and accurate scoring.

### 3.2.1 User Input Validation

Input validation ensures only valid responses (A-D, exit, quit) are accepted, preventing crashes and enhancing user experience.

## 3.3 Design of the Project Software/Algorithm

### 3.3.1 Question Shuffling Logic

The Fisher-Yates shuffle algorithm is used to randomize questions, ensuring unbiased selection:

for (int i = Total\_Questions - 1; i > 0; --i) { int j = rand() % (i + 1);

int temp = questionIndices[i]; questionIndices[i] = questionIndices[j];

questionIndices[j] = temp;

}

### 3.3.2 Scoring System

Each correct answer awards 3 points, with a total possible score of 15 for 5 questions. The score is calculated as:

Score = CorrectAnswerCount × ScorePerQuestion

## 3.4 Implementation of Features

### 3.4.1 Timer and Performance Metrics

A timer tracks quiz duration using steady\_clock:

startTime = steady\_clock::now();

endTime = steady\_clock::now();

int elapsedTime = duration\_cast<seconds>(endTime - startTime).count();

### 3.4.2 User History Tracking

User responses and results are stored in a text file named <username>\_history.txt, ensuring persistent data storage.

## 3.5 Summary

The design and implementation focus on modularity, user interaction, and data persistence, resulting in a robust quiz system.

Chapter 4

# TOOLS AND TECHNIQUES

## 4.1 Standard PC/Laptop

The project was developed on a standard Windows PC with the following specifications:

* OS: Windows 7/8/10/11
* Processor: Any Intel or AMD CPU
* RAM: Minimum 500mb GB

## 4.2 Tools

### 4.2.1 C++ Compiler (GCC/MSVC)

The GNU Compiler Collection (GCC) and Microsoft Visual C++ (MSVC) were used to compile the C++ code, ensuring compatibility and performance [8].

### 4.2.2 Visual Studio Code

Visual Studio Code served as the primary IDE, offering features like code completion and debugging [9].

### 4.2.3 Windows API (for Console Features)

The Windows API enhanced the console interface with color output and font size adjustments

[6].

## 4.3 Summary

The tools and techniques ensured efficient development and a user-friendly interface, leveraging standard C++ libraries and Windows-specific features.

Bibliography

1. Tanenbaum, A. S., "Operating Systems: Design and Implementation," Prentice Hall, 2006.
2. Smith, J., "Educational Quiz Systems," Journal of Educational Technology, 2018.
3. Stroustrup, B., "The C++ Programming Language," 4th Edition, Addison-Wesley, 2013.
4. Booch, G., "Object-Oriented Analysis and Design," Addison-Wesley, 1994.
5. Kernighan, B. W., "The C Programming Language," Prentice Hall, 1988.
6. Microsoft, "Windows API Documentation," 2023.
7. Knuth, D. E., "The Art of Computer Programming, Volume 2," Addison-Wesley, 1998.
8. Stallman, R., "Using the GNU Compiler Collection," Free Software Foundation, 2022.
9. Microsoft, "Visual Studio Code Documentation," 2023.

Appendix A

# Source Code Snippets

A.1 QuizManager Class Definition

class QuizManager

{

private:

const static int Total\_Questions = 5;

const static int Total\_Options = 4;

string Question[Total\_Questions];

string Option[Total\_Questions][Total\_Options];

string Correct[Total\_Questions][1];

string User\_Name;

int SCORE\_ = 0;

int CorrectAnswerCount\_ = 0;

int WrongAnswerCount\_ = 0;

const int Score\_Per\_Question = 3;

const int Total\_Score = NumberOfQuestionsToAsk \* Score\_Per\_Question;

steady\_clock::time\_point startTime;

steady\_clock::time\_point endTime;

public:

QuizManager()

{

LoadQuestions();

ShuffleQuestion();

SetZero();

}

};

Appendix B

# Sample User History File

-------------------------------------------

User Name: test

Question 1: What is the output of the following code: std::cout << 1 + 1; ?

A: 1

B: 2

C: 11

D: 3

User Answer: A

Correct Answer: B

test Result

Time Elapsed: 6 Seconds

Score: 0

Score Percentage: 0%

Correct Answer: 0

Wrong Answer: 1

Percentage Correct: 0%

Result Status: Fail

-------------------------------------------