

PROJECT ANALYSIS —

Quiz Game

Project Overview

C++ Console Application

 C++ Programming

 Console UI

 Interactive Game

 **Mini Project**
Educational Application

 February 2026

Table of Contents

01



Project Overview

02



Code Structure

03



Core Functionality

04



User Interface

05



Scoring System

06



Code Analysis

07



Key Features

08



Future Enhancements

— PROJECT OVERVIEW

Quiz Game

Application Type

**Console-Based
Interactive Quiz**

KEY FUNCTIONS

 **Multiple-Choice Questions**

 **Real-time Score Tracking**

 **Instant Feedback System**

-

OBJECTIVE Educational Tool

Test user knowledge across various subjects including geography, programming, and technology concepts.

-

SCOPE Current Implementation

Features 3 sample questions with 4 options each, totaling 30 possible points, providing a complete quiz experience.



Data Structures & Functions

CORE DATA STRUCTURE

struct Question

```
string question // Question text  
string options[4] // A, B, C, D choices  
char correctAnswer // 'A', 'B', 'C', 'D'  
int points // Question value
```

Design Philosophy

Modular structure with clear separation between data (Question struct) and logic (startQuiz function).

FUNCTION DEFINITIONS

```
void startQuiz(Question q[], int size) {  
    int score = 0;  
    for (int i = 0; i < size; i++) {  
        // Display question & options  
        if (userAnswer == correct) {  
            score += points;  
        }  
    }  
}
```



startQuiz()

Main game logic loop

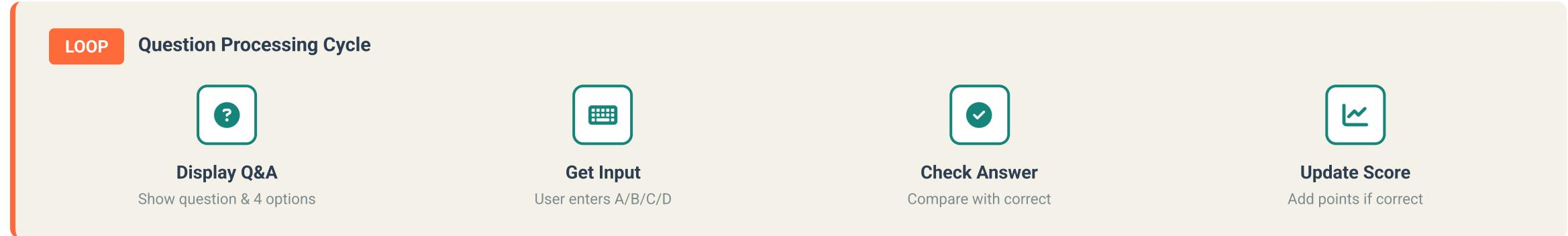
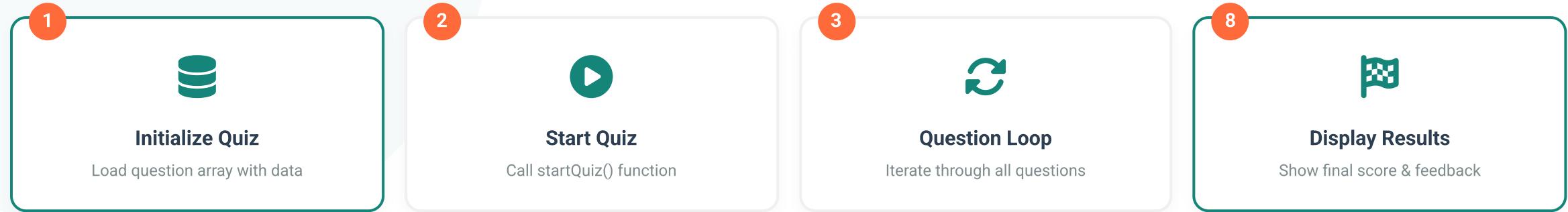


main()

Initialize & start quiz

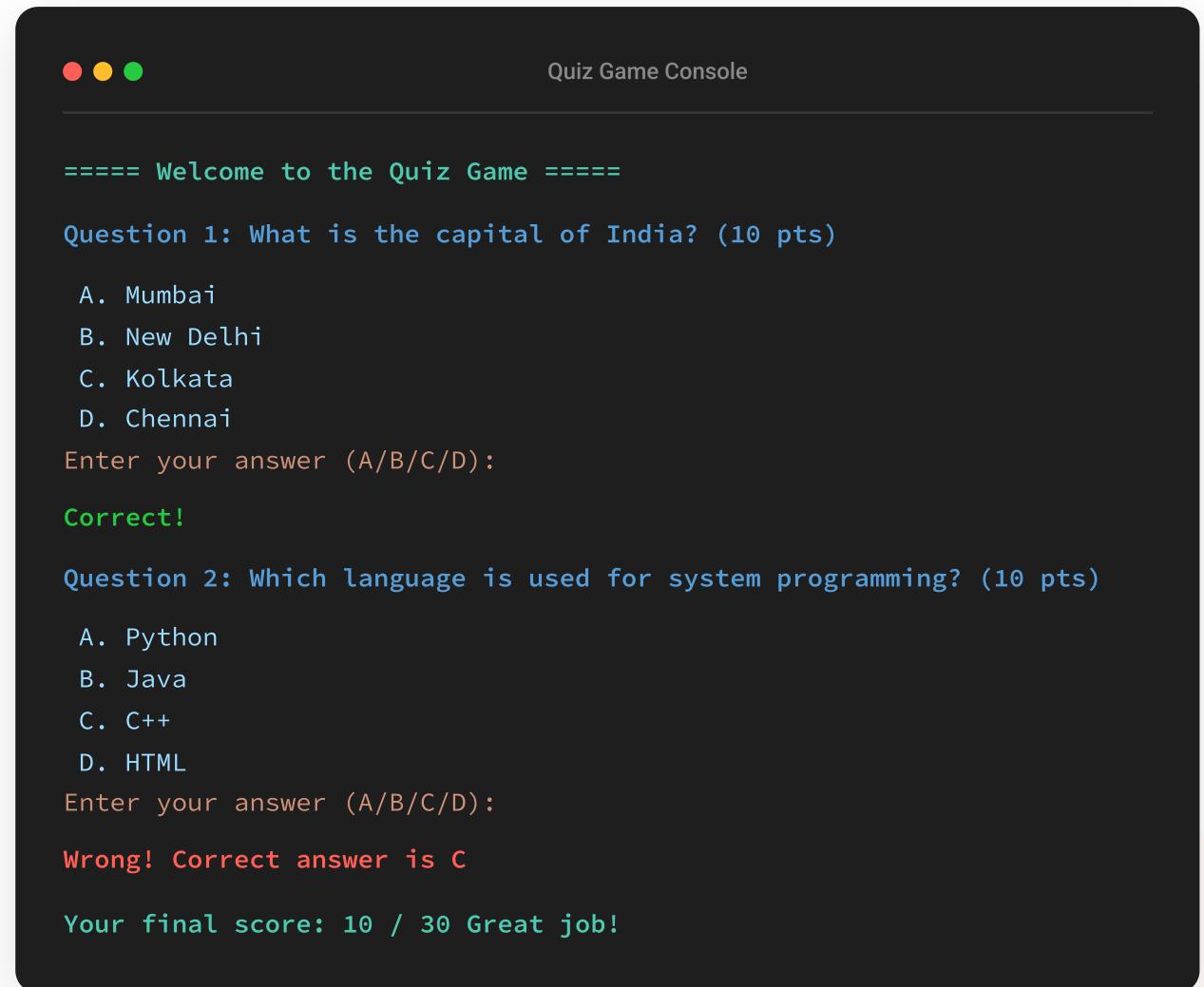
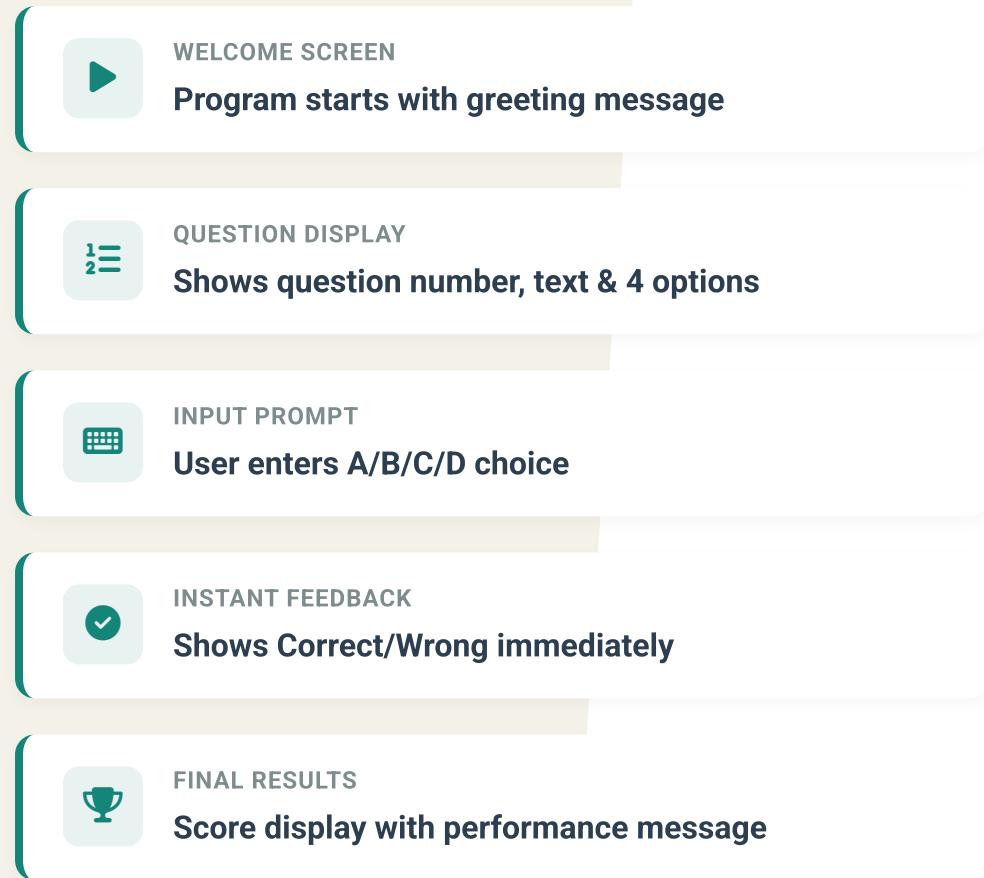
CORE FUNCTIONALITY

Program Flow



— USER INTERFACE

Console-Based Experience



— SCORING SYSTEM

Score Calculation & Feedback

SCORING MECHANISM

+ Points per Question

Variable (10 pts each)

⟳ Accumulation

Score += points (if correct)

-Calculator Total Calculation

Sum of all questions

PERFORMANCE FEEDBACK

≥ 20 pts "Great job!"

< 20 pts "Better luck next time."



Final Score Display

20 / 30

Great job!

💡 Scoring Logic

score starts at 0, increments only for correct answers. Total possible = sum of all question points.

— FUTURE ENHANCEMENTS

Roadmap for Improvement



CORE

Random Question Order

Shuffle questions dynamically to enhance replayability and prevent memorization.



Question Bank Expansion

Expand question database with hundreds of questions across multiple categories.



Difficulty Levels

Implement easy, medium, and hard difficulty tiers with adaptive scoring.



Time Limits

Add countdown timers per question to increase challenge and excitement.



Persistent Score Tracking

Store high scores and progress using file I/O for long-term statistics.



GUI Implementation

Develop graphical user interface using Qt, SFML, or web technologies.

