



SRM
UNIVERSITY AP
Andhra Pradesh

Mini Quiz Game with Scores



Introduction

This project is a console-based Mini Quiz Game developed using C++ (Functions + Arrays + Score Handling).

It allows users to attempt multiple-choice questions and receive a score at the end.

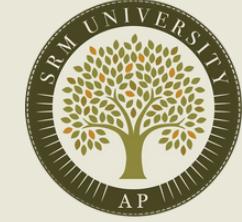
The quiz game performs the following operations:

- **Display questions**
- **Accept user choices**
- **Validate answers**
- **Calculate and display score**

The program demonstrates key C++ concepts—Functions, Arrays, Conditionals, Loops, and Basic Modular Structure.

Project Objectives:

- Implement a functioning quiz game using C++.
- Ask the user a set of multiple-choice questions.
- Validate answers and update score accordingly.
- Display results such as:
 - ✓ Correct / Wrong
 - ✓ Final Score
 - ✓ Performance Message (optional)
- Ensure modularity using separate functions.
- Keep the game simple, clear, and beginner-friendly.





Key System Features:

Quiz Questions

- Stored in arrays or structures
- Each question includes 4 options and 1 correct answer

User Interaction

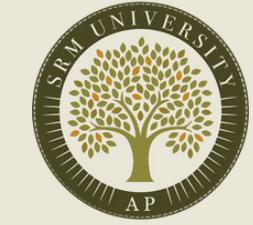
- Enter option (A/B/C/D)
- Input validation

Score System

- Score increases for correct answers
- Shows result per question

Final Output

- Displays final marks
- Provides feedback



System Architecture:

Architecture

User Layer

- User reads questions from console
- Inputs answer choices
- Game Logic Layer

Functions:

- askQuestion()
- checkAnswer()
- calculateScore()

Data Layer

- Arrays to store questions, options, and correct answers



Programming Concepts Used:

Functions

- Separate logic for modularity
- Example: void askQuestion(..)

Arrays/Strings

- Store questions, options, answers

Conditionals

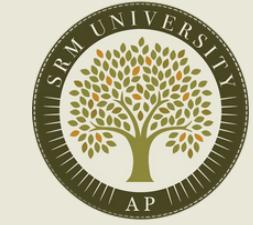
- if(option == correctAnswer)

Loops

- Iterate through all questions

Basic Scoring Mechanism

- Integer counter for score



Workflow Explanation:

Step-by-step process:

1. Game displays the first question.
2. User enters an option (A/B/C/D).
3. Program checks if answer is correct.
4. Score is updated.
5. Next question appears.
6. After final question, score is calculated.
7. Result is displayed with feedback.
8. Option to restart (optional).



Advantages of the System:

1. Simple and easy to understand.
2. Demonstrates essential C++ concepts.
3. Easily extendable (more questions, difficulty levels).
4. Great for academic mini-project.
5. No external files needed.
6. Beginner-friendly implementation.

Code Implementation Overview:

The quiz game is implemented using:

- Arrays for questions
- Functions for structure
- Conditional logic for scoring
- Loops to iterate over questions

Core functions implemented:

- askQuestion()
- checkAnswer()
- main()



SRM
UNIVERSITY AP
Andhra Pradesh

Example Code Snippet:

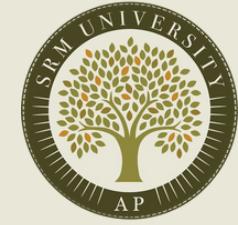


SRM
UNIVERSITY AP
Andhra Pradesh

```
void askQuestion(string q, string a, string b, string c, string d, char correct, int &score) {  
    cout << q << endl;  
    cout << "A. " << a << endl;  
    cout << "B. " << b << endl;  
    cout << "C. " << c << endl;  
    cout << "D. " << d << endl;  
  
    char userAnswer;  
    cout << "Your answer: ";  
    cin >> userAnswer;  
  
    if (toupper(userAnswer) == correct) {  
        cout << "Correct!\n";  
        score++;  
    } else {  
        cout << "Wrong! Correct answer: " << correct << "\n";  
    }  
}
```



Sample Output:



SRM
UNIVERSITY AP
Andhra Pradesh

Question 1: Capital of France?

- A. Paris
- B. London
- C. Rome
- D. Berlin

Your answer: A

Correct!

Question 2: $2 + 2$?

- A. 3
- B. 4
- C. 5
- D. 6

Your answer: B

Correct!

Final score: 2 / 2

Excellent performance!

Conclusion:

- This project successfully implements a functional quiz game using C++.
- Demonstrates use of loops, conditionals, arrays, and functions.
- Offers a clean and structured design.
- Simple, lightweight, and ideal for C++ beginners.
- Can be expanded easily with more questions or features.

