=> Technical Writing: Understanding the Programming MCQ Quiz Program <=

This Java program is a Graphical User Interface (GUI) based quiz application built using Swing and AWT libraries. It allows users to test their knowledge of programming languages (Java, Python, and C++) through multiple-choice questions (MCQs). Below is a structured explanation of how the program works.

=> Purpose of the Program:

-> Provide an interactive quiz experience for learners.
-> Allow users to select a programming language (Java, Python, or C++).
-> Present multiple-choice questions with four options.
-> Track score, attempted questions, and wrong answers.
-> Display final results and explanations for incorrect answers.

=> Core Components:

1. Frame & Panels

-> JFrame: The main application window.
-> JPanel with CardLayout: Manages multiple screens (Welcome, Quiz, Result, Explanation).

2. Data Storage

-> Questions are stored in 2D arrays (javaQuestions, pythonQuestions, cppQuestions).

-> Each question contains:

* Question text
* Four options
* Correct answer (A/B/C/D)
* Explanation

3. Quiz Flow Variables

-> currentQuestion: Tracks which question is being displayed.
-> score: Counts correct answers.
-> attempted: Counts total attempted questions.
-> wrongAnswers: Stores incorrectly answered questions for later review.

=> Program Flow:

1. Welcome Screen

-> Displays a greeting message.
-> Provides buttons to choose a language (Java, Python, C++).
-> Clicking a button starts the quiz for that language.

2. Quiz Screen

-> Shows the current question (questionLabel).
-> Displays four options (JRadioButton).
-> User selects an option and clicks Next.
-> Answer is checked:

* If correct → score increases.
* If wrong → question is added to wrongAnswers.

3. Result Screen

-> Shows:

* Total questions attempted
* Correct answers
* Percentage score

-> Provides a button to view explanations.

4. Explanation Screen

-> Displays explanations for all wrong answers.
-> If all answers are correct, shows a congratulatory message.

=> Key Methods:

-> createAndShowGUI()
Initializes the frame, sets up panels, and makes the GUI visible.

-> createWelcomePanel()
Builds the welcome screen with language selection buttons.

-> createQuizPanel()
Builds the quiz interface with question label, options, and next button.

-> startQuiz(String[][] quizData)
Resets quiz progress and loads the first question.

-> loadQuestion()
Displays the current question and options.

-> checkAnswer()
Validates the selected answer, updates score, and moves to the next question.

-> showFinalScore()
Displays the final result after all questions are attempted.

-> showExplanations()
Shows detailed explanations for wrong answers.

=> How It Works Step-by-Step:

-> User opens the program → Welcome screen appears.
-> User selects a language → Quiz starts.
-> Each question is displayed with four options.
-> User selects an answer → Program checks correctness.
-> After all questions → Result screen shows performance.
-> User can view explanations for mistakes.

=> Advantages of This Program:

-> Simple and user-friendly interface.
-> Supports multiple programming languages.
-> Provides immediate feedback with explanations.
-> Encourages learning by reviewing mistakes.

=> Possible Enhancements:

-> Add more questions dynamically from a file or database.
-> Include a timer for each question.
-> Provide difficulty levels (easy, medium, hard).
-> Track progress across multiple quiz sessions.