Technical Description - C++ Quiz Game Project

1. Project Overview and Task Division

This C++ project implements a terminal-based quiz game using object-oriented principles.

Key features:

- Interactive menu for playing the quiz, viewing leaderboard, and user history
- Uses command-line arguments for controlling whether the 50/50 lifeline is allowed and for adding new questions to the question bank
- Stores user scores in leaderboard.txt and game session summary in output.txt

Team Members:

- Student A
- Student B

Task Division:

- Student A:
  - Created Question and LeaderboardEntry classes
  - Implemented quiz question flow and 50/50 logic
- Student B:
  - Developed command-line interface for toggling 50/50 and adding questions
  - Handled file I/O for reading questions and updating scores

2. Data Structures Used

- Question: Holds the question string, 4 options, and the index of the correct answer
- LeaderboardEntry: Stores player name and score for leaderboard tracking
- QuizGame: Core class managing question flow, scoring, file handling, and user interface

3. File Communication and Structures

- questions.txt: Plain text file storing all questions in blocks of 6 lines (1 question, 4 options, 1 correct index)
- leaderboard.txt: Appends scores and is later sorted in-game to reflect latest standings
- output.txt: Stores a formatted session summary with username, score, and timestamp

4. Commands and Configuration via Arguments

Technical Description - C++ Quiz Game Project

The project uses command-line arguments to modify behavior and extend functionality:

1. Add a question:

```
./quiz_game add "What is 2+2?" "1" "2" "3" "4" 2
```

Appends a new question to questions.txt with options and correct answer index

2. Enable or disable 50/50 lifeline:

```
./quiz_game true     (default, enables 50/50)
./quiz_game false    (disables 50/50)
```

If no arguments are provided, the game launches the default interactive menu.