A Project Report

On

**“Implementation of Cricket Game Simulation”**

For The Course

**“Project-I (ICT-1200)”**

**By**

Kuldip Saha Mugdha (IT-21018)

**Supervised by**

**Dr. Monir Morshed Sir**

Professor

Department of ICT, MBSTU.



**DEPARTMENTOFINFORMATIONANDCOMMUNICATIONTECHNOLOGY**

**MAWLANABHASHANISCIENCEANDTECHNOLOGYUNIVERSITY**

SANTOSH, TANGAIL-1902, Dhaka, Bangladesh

**Contents**

**Page No.**

**Cover Page----------------------------------------------------------------- i**

**Declaration---------------------------------------------------------------- iv**

**Acknowledgements------------------------------------------------------- v**

**Chapter 1: Introduction of Project**

1.1. Project Overview----------------------------------------------------------- 6

1.2. Purpose of The Project----------------------------------------------------------- 6

1.3. Project Scope --------------------------------------------------------------------- 7

1.4. Hardware and Software Requirements ---------------------------------------- 7

**Chapter - 2: Design & Development**

2.1. Project Diagram ------------------------------------------------------------------ 9

2.2. Uses Function and Details ------------------------------------------------------ 11

2.3. Output and description ---------------------------------------------------------- 12-15

**Chapter - 3: Limitation & Future Scope**

3.1. Limitation ---------------------------------------------------------------------------- 16

3.2. Future Scope -------------------------------------------------------------------------- 18

**Chapter - 4: Conclusion**

4.1. Conclusion -------------------------------------------------------------------------- 21

**Declaration**

This is to certify that the work presented in this project is carried out by the candidate under the supervision of **Monir Morshed Sir** in the department of Information and Communication Technology, MBSTU, Tangail, Bangladesh. It is also declared that neither of this project has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Signature of Supervisor

**Monir Morshed Sir**

Professor

Dept. of ICT, MBSTU

**Acknowledgements**

I am immensely grateful to the Almighty God for the successful completion of my dissertation. I wish to begin by extending my heartfelt gratitude to my supervisor, Monir Morshed Sir, Professor in the Department of Information and Communication Technology at MBSTU. His unwavering support paved the way for the continuation of my Project-I.

I would also like to express my deep appreciation for my supervisor's invaluable guidance, insightful suggestions, constant encouragement, and unwavering support throughout the entirety of the project. However, words seem insufficient to truly acknowledge the tremendous effort put forth by our esteemed teacher.

As always, our gratitude towards our family knows no bounds. The unwavering love and support from our parents continue to be the cornerstone of our lives.

**Chapter-01 Introduction**

# 1.1.Project Overview:

Cricket, a widely popular sport, enthralls millions around the globe with its strategic gameplay and thrilling moments. In an effort to bring the excitement of cricket to the digital realm, this project introduces a Cricket Game Simulation implemented in C++.

The project aims to simulate a simplified version of a cricket match, providing users with an immersive experience of playing and strategizing through the game. Utilizing fundamental concepts of C++ programming, the simulation allows users to select teams, participate in a virtual coin toss, and simulate the progress of a cricket match ball by ball.

* **Team Selection:** Users can choose their favorite teams from a list of available options.
* **Coin Toss:** A virtual coin toss determines which team bats first, adding an element of randomness to the game.
* **Innings Simulation:** The program simulates the progress of each team's innings, allowing users to input the outcome of each ball, including runs scored, wickets taken, wides, and no balls.
* **Feedback and Scoreboard:** Users receive real-time feedback on the match progress, including the current score, wickets fallen, and required run rate.
* **Result Determination:** At the conclusion of both innings, the program determines the winner based on the total runs scored by each team.

# 1.2.Purpose of The Project:

The purpose of the Cricket Game Simulation Project is to provide users with an entertaining and educational experience of playing cricket in a simulated environment. Through interactive gameplay, the project aims to familiarize users with the rules and dynamics of cricket, while also promoting skill development, community engagement, and creativity. By combining entertainment with learning opportunities, the project seeks to engage users of all ages in the exciting world of cricket, fostering a love for the sport and promoting programming skills in the process.

# 1.3.Project Scope:

The project scope encompasses the development of a Cricket Game Simulation implemented in C++. It includes features such as team selection, coin toss simulation, innings progression with user-inputted ball outcomes, real-time feedback on match progress, and result determination based on total runs scored. The simulation aims to provide users with an immersive experience of playing cricket in a virtual environment while promoting entertainment, education, skill development, and community engagement. The project focuses on delivering a user-friendly and engaging gaming experience within the constraints of a command-line interface, with potential for future enhancements and expansions to include additional features and functionalities.

# 1.4. Hardware and Software Requirement:

**Hardware Requirements:**

1. Computer or laptop with a minimum of 2 GHz processor.
2. At least 2 GB of RAM to ensure smooth execution of the program.
3. Sufficient storage space for storing the program files and any additional resources.

**Software Requirements:**

1. **Operating System:** Compatible with Windows, macOS, or Linux.
2. **C++ Compiler:** Required for compiling and running the C++ code.
3. **Popular options include:**

* GCC (GNU Compiler Collection)
* Visual C++ Compiler (for Windows)
* Clang (C/C++ compiler)

1. **Integrated Development Environment (IDE):** While not mandatory, an IDE can streamline the development process. Recommended IDEs include:

* Visual Studio Code
* Code::Blocks
* Dev-C++

1. **Terminal or Command Prompt:** Necessary for executing the compiled program and interacting with the command-line interface.
2. **Text Editor:** Users may opt to use a text editor for viewing and editing the source code files. Popular text editors include:

* Notepad++ (Windows)
* Sublime Text
* Atom

**Optional:**

1. **Git:** Version control system for managing code revisions and collaboration (optional but recommended for team-based development).

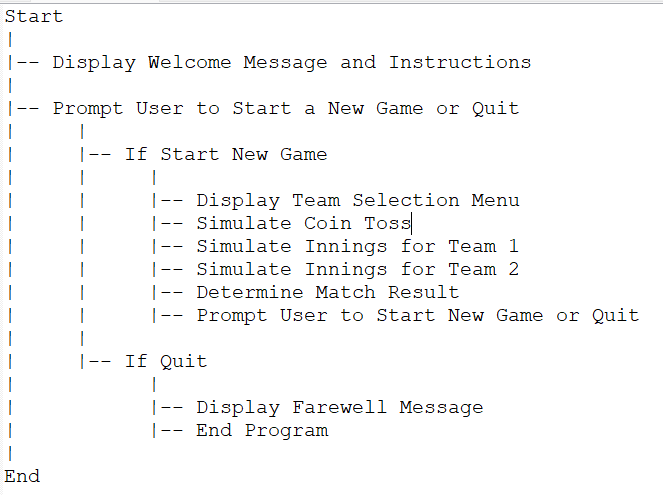
**GitHub:** Online platform for hosting and sharing code repositories (optional but useful for collaborative projects).

These hardware and software requirements are essential for developing and running the Cricket Game Simulation project. Users should ensure compatibility with their system specifications and install the necessary software components to facilitate smooth execution and development.

**Chapter-02**

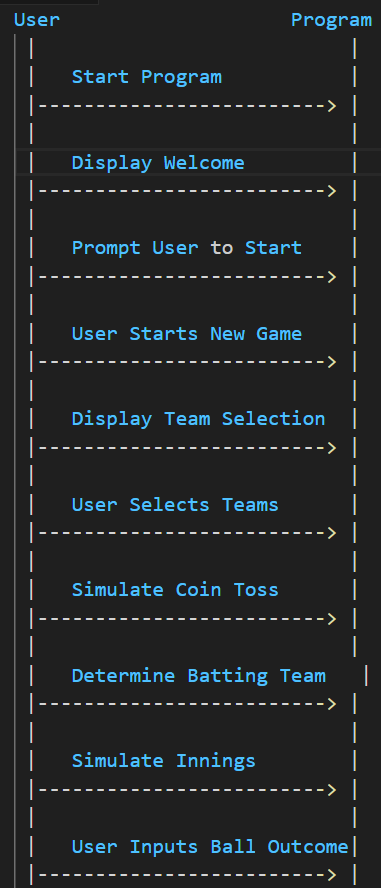
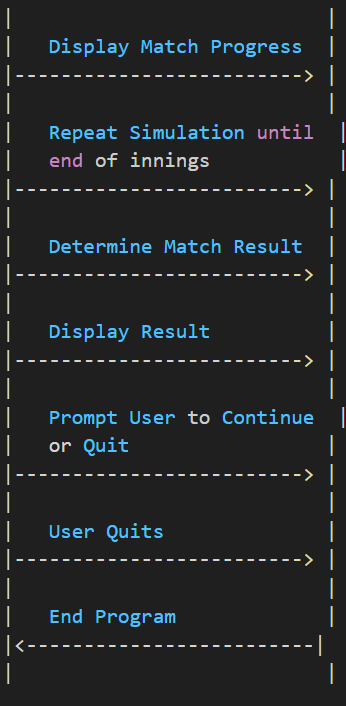
**Design and Development**

# 2.1. Project Diagram:



**Project Sequence Diagram:**

Since the project primarily consists of a command-line interface and sequential execution of functions, a traditional sequence diagram might not be the most suitable representation. However, we can illustrate the sequence of interactions between the user and the program through a simplified sequence diagram:



**2.2.Uses Function and Details**

1. **‘main()’:**

* This is the entry point of the program.
* It displays a welcome message and instructions to the user.
* Allows the user to start a new game or quit the program.
* Handles team selection, coin toss simulation, and match simulation.
* Determines the winner of the match and prompts the user for further action.

1. **‘teamchoose(int x)’:**

* This function takes an integer input representing the team choice.
* Returns the corresponding team name based on the input.
* Each integer input corresponds to a specific team name (e.g., 1 for "Batch15", 2 for "Batch16", etc.).

1. **‘game():’**

* Simulates the innings of one team.
* Allows the user to input the outcome of each ball (runs, wickets, wides, no balls).
* Tracks the total runs scored and the number of wickets fallen.
* Provides feedback on ball outcomes (e.g., clean bowl, catch, LBW, run out).
* Generates a scoreboard after the innings and calculates the required run rate for the opponent team.

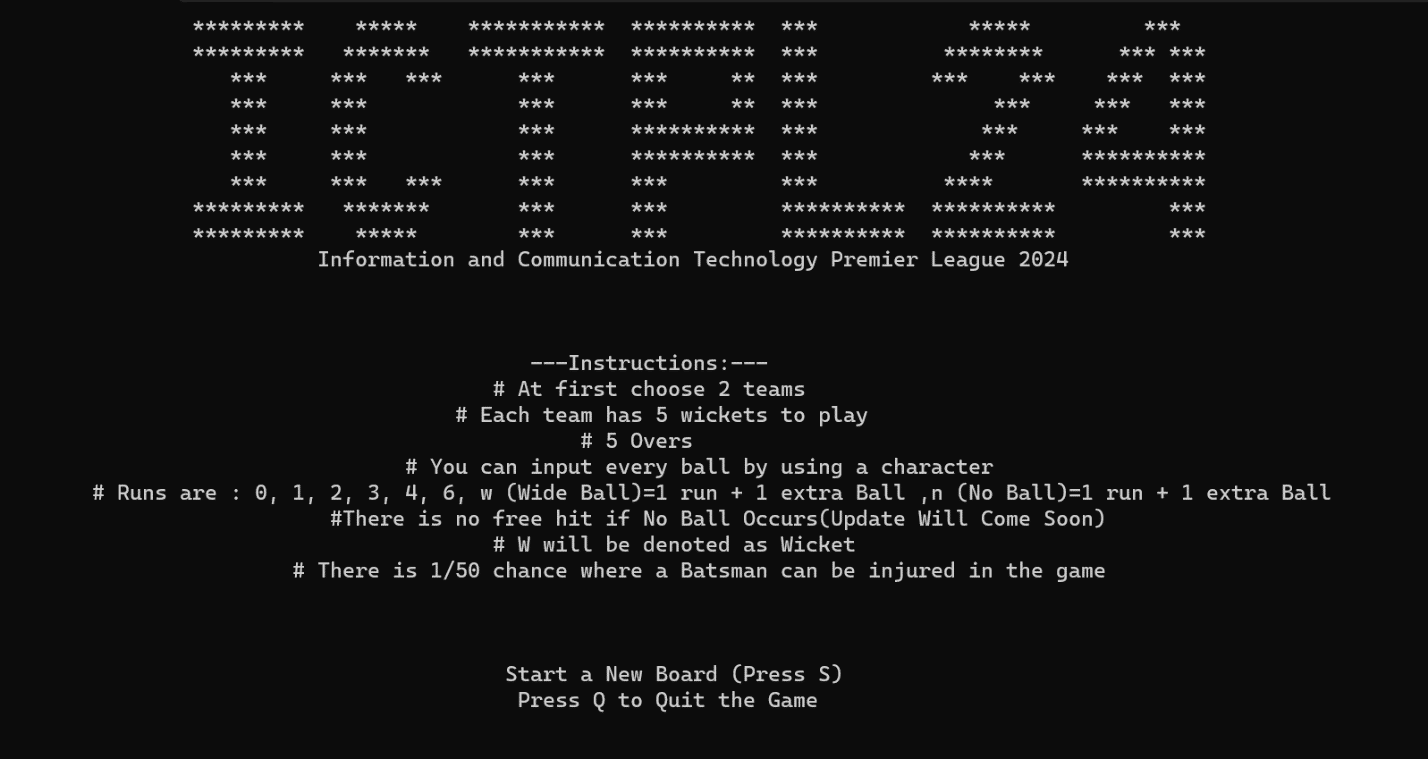
1. **‘game2()’:**

* Similar to game(), but simulates the innings of the opponent team.
* Uses the total runs scored by the first team to determine the target for the opponent team.
* Tracks the progress of the innings and provides feedback on ball outcomes.
* Generates a scoreboard after the innings and determines the winner of the match.

1. **Other utility functions:**

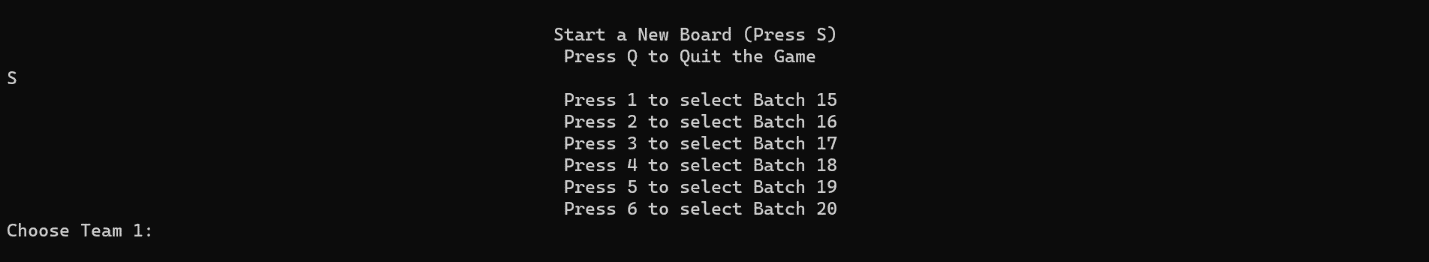
* There might be additional utility functions used for input validation, generating random numbers, or formatting output.

**2.3. Output and description:**

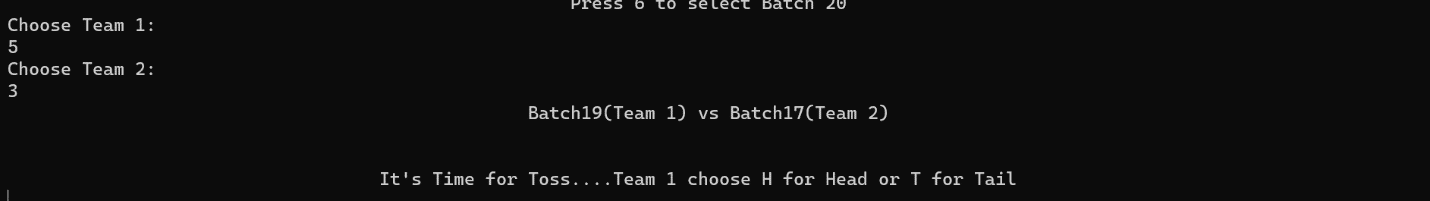
Home Page:

**Main Menu:** The main menu option include a Start button and a Quit button.

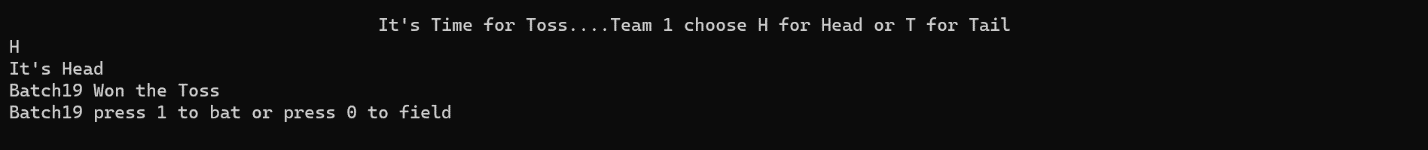
**Start:** Pressing S to start the Board.



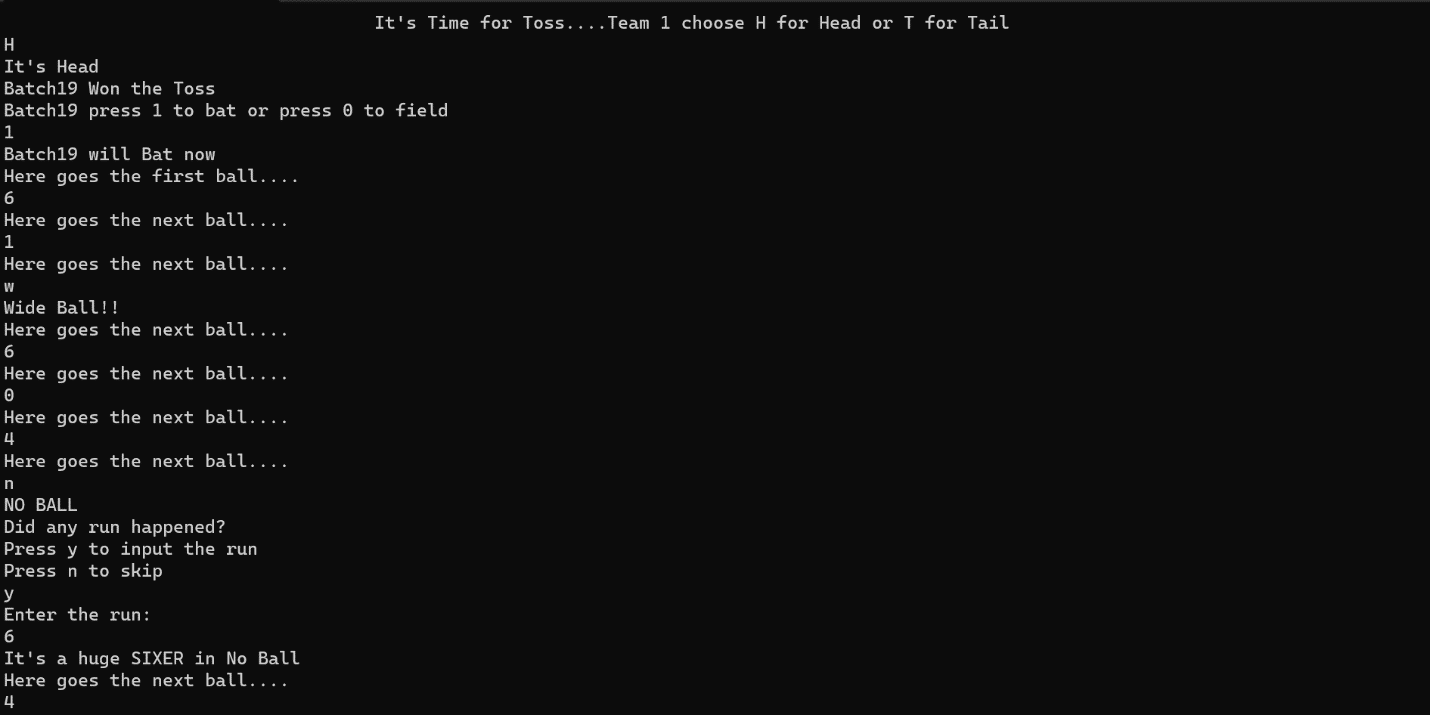
**Choose Team:**

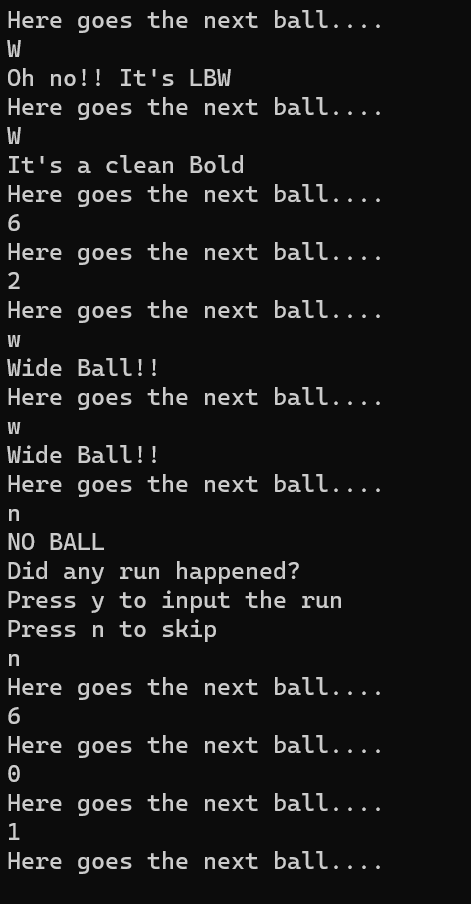


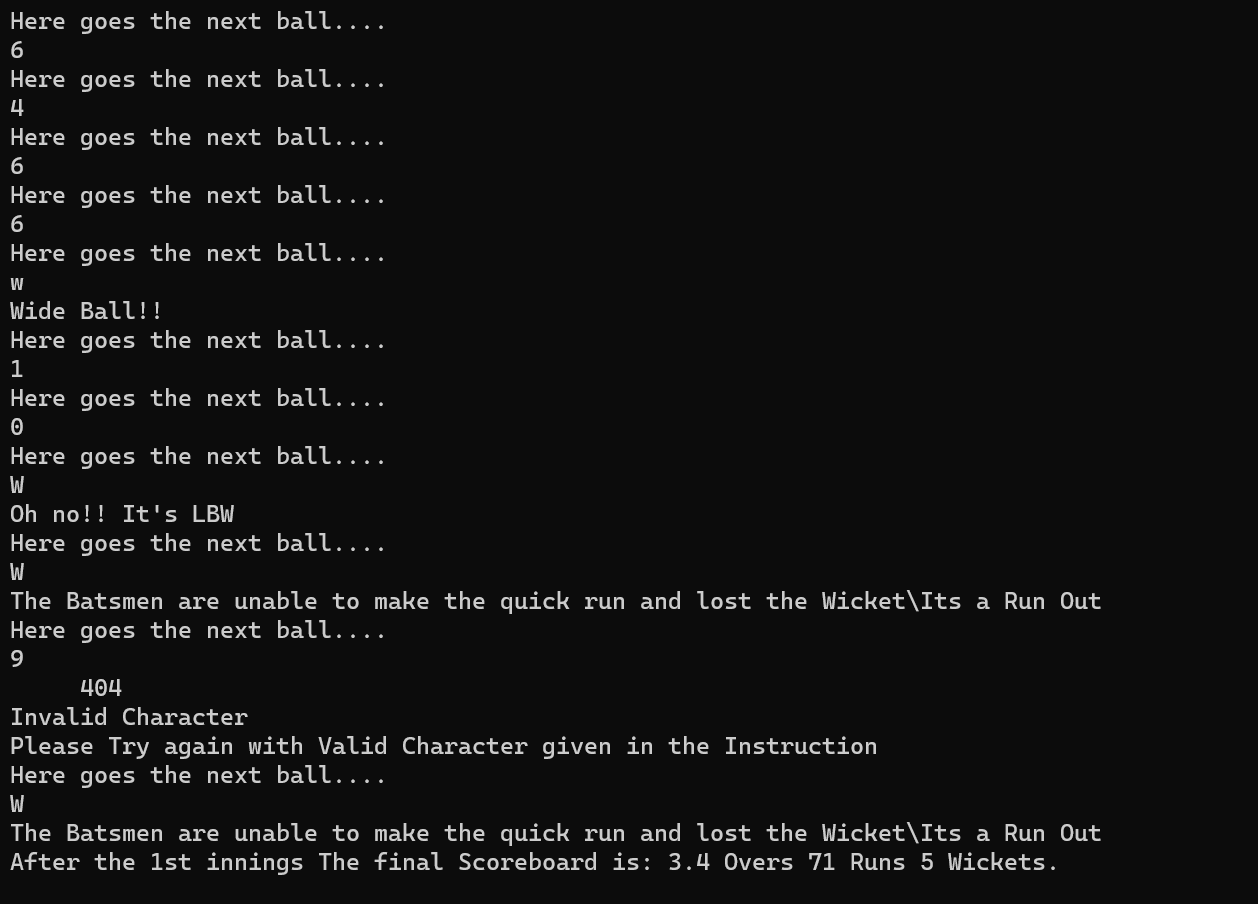
**Toss:**



**1st Innings:**

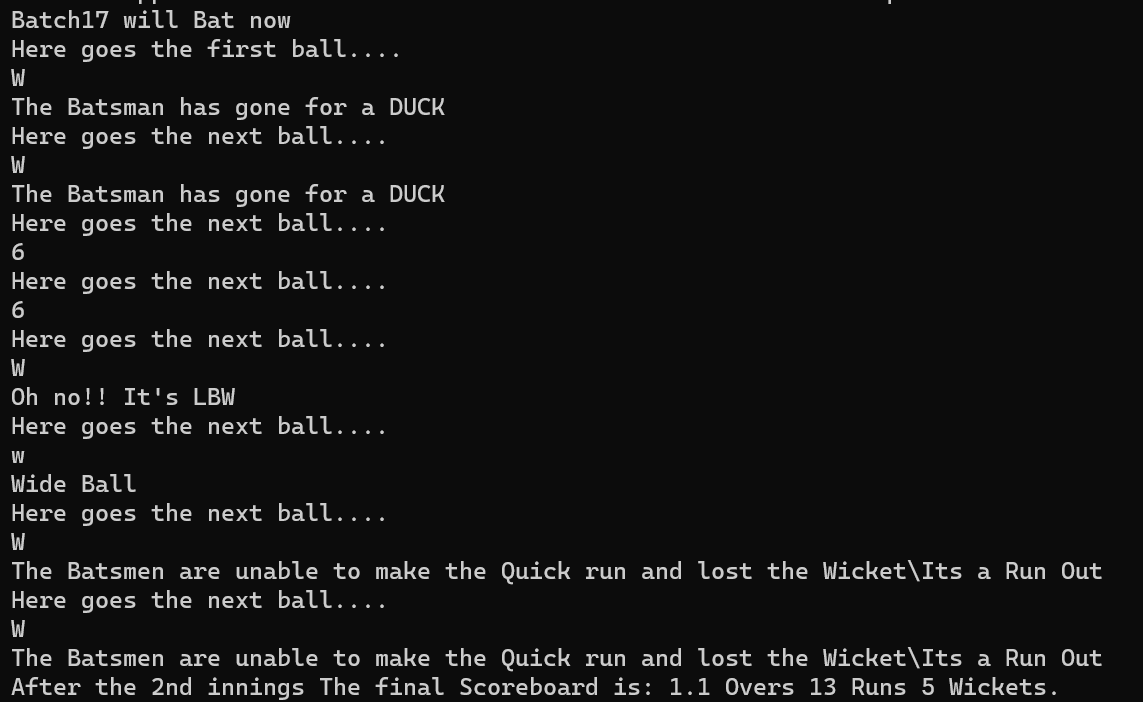
****

****

****

**Target Board:**

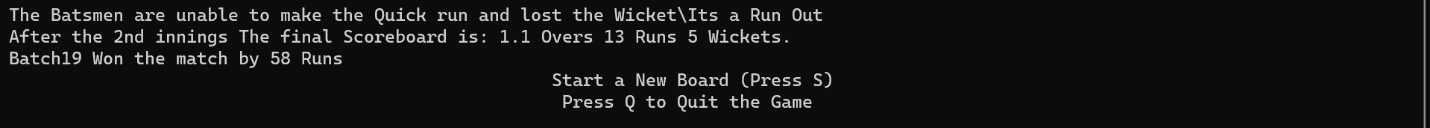
****

**2nd Innings:**

**Final Result:**

****

**Again Start or Quit:**

****

If a operator press Q the console will automatically be closed.

**Chapter-03**

**Limitation and Future Scope**

# 3.1. Limitation

**Command-Line Interface (CLI):**

* The project is limited to a command-line interface, which may not provide the most user-friendly or visually appealing experience compared to graphical user interfaces (GUIs).
* Lack of graphical elements and interactivity may limit the engagement level, especially for users accustomed to modern GUI applications.

**Simplistic Gameplay:**

* The gameplay mechanics are relatively simplistic and may not fully capture the complexity and nuances of real cricket matches.
* Limited options for user interaction, such as inputting ball outcomes, may restrict the depth of strategic decision-making and gameplay variability.

**Limited Features:**

* The project lacks advanced features commonly found in commercial cricket simulation games, such as player customization, team management, career modes, and multiplayer functionality.
* Lack of diversity in gameplay modes and scenarios may reduce long-term replay value and user engagement.

**Input Limitations:**

* User input for simulating ball outcomes is restricted to specific characters (e.g., '0', '1', '2', '3', '4', '6', 'W', 'w', 'n'), limiting the flexibility and realism of gameplay.
* Lack of support for mouse input or graphical interfaces may hinder accessibility for users unfamiliar with command-line navigation.

**Randomness and Predictability:**

* The project's random number generation for simulating ball outcomes may lead to unpredictable or unrealistic gameplay scenarios.
* Limited control over the randomness of outcomes may result in repetitive or unbalanced gameplay experiences.

**Scalability and Extensibility:**

* The project may face challenges in scaling up or accommodating additional features, teams, or game modes without significant modifications to the codebase.
* Lack of modularity and abstraction may hinder code maintenance and future expansion efforts.

**Platform Dependency:**

* The project's reliance on specific compilers, libraries, or system configurations may introduce platform dependencies and compatibility issues across different operating systems or development environments.

# 3.2. Future Scope

The Cricket Game Simulation project lays the groundwork for a command-line-based cricket game. While the current implementation is functional, there are several avenues for future enhancements and expansions to make the project more comprehensive and engaging. Here are some potential future scope areas:

**Graphical User Interface (GUI):**

* Develop a graphical user interface to replace the command-line interface, providing users with a more visually appealing and intuitive experience.
* Implement graphics, buttons, and interactive elements to enhance user engagement.

**Multiplayer Mode:**

* Introduce multiplayer functionality to allow users to play against each other in real-time or take turns simulating innings.
* Implement network communication to enable online multiplayer matches.

**Team and Player Customization:**

* Allow users to customize their teams, including team names, player names, and player attributes (batting skill, bowling skill, etc.).
* Implement a player database to store and retrieve player statistics.

**Advanced Gameplay Features:**

* Introduce more advanced gameplay features, such as different match formats (Test, One Day, T20), power plays, and strategic decisions for users to make during the game.
* Add a career mode where users can manage their team across multiple matches or seasons.

**Improved Randomization and Realism:**

* Refine the random number generation algorithm to ensure more realistic and balanced gameplay outcomes.
* Introduce more dynamic and context-aware commentary to enhance the realism of the simulation.

**Enhanced User Feedback:**

* Provide detailed and visually appealing feedback during gameplay, including animations, scorecards, and highlights.
* Implement sound effects and commentary to further immerse users in the virtual cricket experience.

**Tournament Mode:**

* Create a tournament mode where users can simulate entire cricket tournaments with multiple teams and matches.
* Implement knockout stages, league tables, and a trophy presentation ceremony.

**Save and Load Functionality:**

* Allow users to save their progress during a game and resume it later.
* Implement a save/load mechanism to store game states and statistics.

**Platform Compatibility:**

* Ensure compatibility with a wider range of platforms, including Windows, macOS, and Linux.
* Provide detailed installation instructions and support for different compilers.

**Code Refactoring and Modularity:**

* Refactor the codebase to enhance modularity and maintainability.
* Break down the code into separate functions, classes, or modules to improve readability and facilitate future updates.

By addressing these future scope areas, the Cricket Game Simulation project can evolve into a more comprehensive and feature-rich cricket simulation, offering users a broader and more immersive gaming experience.

**Chapter-05**

**Conclusion**

## **5.1. Conclusion**

In conclusion, the Cricket Game Simulation project provides a foundation for a command-line-based cricket game, offering users an interactive and entertaining experience within a simulated cricket match. While the current implementation has its limitations, including a simplistic gameplay model and a command-line interface, it serves as a valuable learning opportunity for programming enthusiasts interested in game development, algorithm design, and user interface design.

Despite its limitations, the project demonstrates key concepts in software development, including input/output handling, conditional logic, function implementation, and random number generation. It also introduces users to the rules and dynamics of cricket, fostering a deeper understanding and appreciation for the sport.

Looking ahead, there is significant potential for future enhancements and expansions to make the project more comprehensive and engaging. Possible future scope areas include the development of a graphical user interface, multiplayer functionality, team and player customization options, advanced gameplay features, and tournament modes.

Overall, the Cricket Game Simulation project highlights the creativity, problem-solving skills, and technical proficiency of its developers. With continued iteration and refinement, it has the potential to evolve into a fully-featured cricket simulation game that captures the excitement and complexity of real-world cricket matches.