

Advanced Math Quiz Game

Submitted By

Student Name	Student ID
Md Junayed Bin Karim	0242220005101667
TITAS SARKER	0242220005101864

LAB PROJECT REPORT

This Report Presented in Partial Fulfillment of the course **CSE324:
Operating System Lab in the Computer Science and Engineering
Department**



DAFFODIL INTERNATIONAL UNIVERSITY

Dhaka, Bangladesh

17 April, 2025

DECLARATION

We hereby declare that this lab project has been done by us under the supervision of **Ibrahim Patwary Khokan, Lecturer**, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere as lab projects.

Submitted To:

Ibrahim Patwary Khokan

Lecturer

Department of Computer Science and Engineering(CSE) Daffodil
International University

Submitted by

Junayed Bin Karim
0242220005101667
Dept. of CSE, DIU

COURSE & PROGRAM OUTCOME

The following course have course outcomes as following:

Table 1: Course Outcome Statements

CO's	Statements
CO1	Demonstrate proficiency in shell scripting and command-line operations in Unix-based operating systems.
CO2	Design and implement interactive, user-driven applications that leverage operating system functionalities.
CO3	Apply robust error handling, control flow, and modular programming practices to develop automation solutions.

Table of Contents

Declaration i Course & Program Outcome ii

1 Introduction 1 1.1 Introduction 1 1.2 Motivation 1 1.3 Objectives 1 1.4 Feasibility Study 1
1.5 Gap Analysis 1 1.6 Project Outcome 2

2 Proposed Methodology/Architecture 3 2.1 Requirement Analysis & Design Specification 3
2.1.1 Overview 3 2.1.2 Proposed Methodology/ System Design 3 2.1.3 UI Design 3 2.2
Overall Project Plan 3

3 Implementation and Results 5 3.1 Implementation 5 3.2 Performance Analysis 5 3.3
Results and Discussion 5

4 Engineering Standards and Mapping 6 4.1 Impact on Society, Environment and
Sustainability 6 4.1.1 Impact on Life 6 4.1.2 Impact on Society & Environment 6 4.1.3 Ethical
Aspects 6 4.1.4 Sustainability Plan 7 4.2 Project Management and Team Work 7 4.3 Complex
Engineering Problem 7 **5 Conclusion 8** 5.1 Summary 8 5.2 Limitation 8 5.3 Future Work 8

References 9

Chapter 1

Introduction

1.1 Introduction

The **Advanced Math Quiz Game** is a Bash script-based application designed to test and enhance users' basic and advanced math skills through a progressive set of challenges. The game dynamically adjusts difficulty based on the player's performance, rewards progress, and provides an engaging way to practice mathematical operations.

1.2 Motivation

The motivation behind this project is multifold:

- **Learning Enhancement:** To provide a fun, interactive way for users to practice arithmetic operations.
- **Automation Practice:** To demonstrate the use of shell scripting for automation and interactive user interfaces.
- **Skill Development:** To serve as a practical demonstration of control structures, arithmetic operations, and user I/O in shell scripting.

1.3 Objectives

- Develop an engaging quiz game with dynamic difficulty levels.
- Implement a scoring system that provides level-based rewards.
- Reinforce fundamental Bash scripting techniques including random number generation, conditional branching, and loops.
- Create a maintainable code base that can be extended with additional features or integrated with other systems.

1.4 Feasibility Study

The project was deemed feasible because:

- It leverages simple and effective tools available in most Unix-based systems.
- The Bash language is sufficient for the requirements, including arithmetic operations and user interaction.
- The development process was straightforward, with most operating systems supporting Bash environments.

1.5 Gap Analysis

During the conceptual phase, it was identified that:

- The existing educational tools often lack interactive elements to adapt in real time to a user's learning curve.
- Many math quiz applications focus on static difficulty levels rather than adapting to performance.
- There is a gap in easily deployable quiz games that require minimal external dependencies, achievable via Bash scripting.

1.6 Project Outcome

The project successfully demonstrates:

- A dynamic quiz game that adapts difficulty based on user performance.
- An interactive command-line user interface.
- Effective implementation of core programming constructs in a shell scripting context.
- Foundations for possible future enhancements such as integrating with IoT devices or cloud-based monitoring systems.

Chapter 2

Proposed Methodology

2.1 Requirement Analysis & Design Specification

2.1.1 Overview

The project requirements include:

- A timer (10-second limit) for each question.
- Multiple types of arithmetic problems (addition, subtraction, multiplication, division, modulus, squaring, and parity checking).
- A dynamic scoring system with rewards and level adjustments.

2.1.2 Proposed Methodology/System Design

The project is designed using a modular structure in Bash, which includes:

- **Input Module:** Reads player inputs and name.

- **Question Module:** Generates random math questions based on the current score.
- **Scoring Module:** Updates the score, correct and incorrect counters.
- **Feedback Module:** Offers immediate responses and level-based rewards.
- **Persistence Module:** Logs final scores to a file for future reference.

This modular design helps isolate functionalities, making future maintenance and enhancements easier.

2.1.3 UI Design

Given that this project is terminal-based, the UI is text-driven. Key aspects include:

- Clear prompt messages and progress indicators.
- Timeouts for responses to increase the challenge.
- A simple menu to continue or exit the game after each question.

2.2 Overall Project Plan

- **Phase 1:** Requirements gathering and feasibility study.
- **Phase 2:** Design of the modular architecture and creation of the flowchart.
- **Phase 3:** Implementation in Bash scripting, followed by unit testing.
- **Phase 4:** Performance evaluation and gathering user feedback.
- **Phase 5:** Documentation and final submission.

Chapter 3

Implementation and Results

3. Implementation and Results

3.1 Implementation

The core of the project is implemented in a Bash script, which involves:

- Random generation of numbers and operations.
- Use of conditional constructs to adjust difficulty and evaluate answers.
- Incorporation of a timer using the `read -t` command.

- Continuous feedback through modular functions (`ask_question`, `get_level`, `show_reward`, etc.).

code:

```
#!/bin/bash
```

```
SCORE_FILE="math_quiz_scores.txt"
```

```
get_level() {  
    if [ $score -lt 5 ]; then  
        echo "Level: Beginner"  
    elif [ $score -lt 10 ]; then  
        echo "Level: Intermediate"  
    elif [ $score -lt 20 ]; then  
        echo "Level: Advanced"  
    else  
        echo "Level: Math Wizard"  
    fi  
}
```

```
show_reward() {  
    if [ $score -eq 5 ]; then  
        echo "You're improving....."  
    elif [ $score -eq 10 ]; then  
        echo "You're doing really well.....)"  
    elif [ $score -eq 20 ]; then  
        echo "You're a math genius.....>>"  
    fi  
}
```

```
ask_question() {
```

```
if [ $score -lt 5 ]; then
```

```
    max_num=10
```

```
elif [ $score -lt 10 ]; then
```

```
    max_num=20
```

```
else
```

```
    max_num=50
```

```
fi
```

```
operation=$((RANDOM % 7))
```

```
num1=$((RANDOM % max_num + 1))
```

```
num2=$((RANDOM % max_num + 1))
```

```
case $operation in
```

```
0) correct_answer=$((num1 + num2))
```

```
    question="What is $num1 + $num2?" ;;
```

```
1) correct_answer=$((num1 - num2))
```

```
    question="What is $num1 - $num2?" ;;
```

```
2) correct_answer=$((num1 * num2))
```

```
    question="What is $num1 * $num2?" ;;
```

```
3)
```

```
    while [ $num2 -eq 0 ]; do
```

```
        num2=$((RANDOM % max_num + 1))
```

```
    done
```

```
    correct_answer=$((num1 / num2))
```

```
    question="What is $num1 / $num2? (Integer division)" ;;
```

```
4)
```

```
    while [ $num2 -eq 0 ]; do
```

```
        num2=$((RANDOM % max_num + 1))
```

```
    done
```

```
    correct_answer=$((num1 % num2))
```

```
    question="What is $num1 % $num2? (Remainder)" ;;
```



```

5) correct_answer=$((num1 * num1))

question="What is $num1 squared?" ;;

6)

question="Is $num1 even or odd? (Type 'even' or 'odd')"

echo "$question"

read -t 10 -p "Your answer: " user_answer

if (( num1 % 2 == 0 )); then

    correct="even"

else

    correct="odd"

fi

if [[ "$user_answer" == "$correct" ]]; then

    echo "Correct!"

    return 1

else

    echo "Incorrect! The correct answer was $correct."

    return 0

fi

return ;;

esac

echo "$question"

read -t 10 -p "Your answer: " user_answer

if [[ $? -ne 0 ]]; then

    echo -e "\nTime's up"

    echo "The correct answer was $correct_answer."

    return 0

fi

```

```

if [[ "$user_answer" -eq "$correct_answer" ]]; then

    echo "Correct!"

    return 1

else

    echo "Incorrect..... The correct answer was $correct_answer."

    return 0

fi
}

play_game() {

    score=0

    correct_count=0

    incorrect_count=0

    read -p "Enter your name: " player_name

    while true; do

        ask_question

        result=$?

        if [[ $result -eq 1 ]]; then

            score=$((score + 1))

            correct_count=$((correct_count + 1))

            show_reward

        else

            incorrect_count=$((incorrect_count + 1))

        fi

        echo "Score: $score | Correct: $correct_count | Incorrect: $incorrect_count"

        get_level

        echo "-----"

```

```
read -p "Do you want to continue? (y/n): " continue_game

if [[ "$continue_game" != "y" ]]; then

    echo "Thanks for playing, $player_name!"

    echo "Final Score: $score | Correct: $correct_count | Incorrect: $incorrect_count"

    get_level

    echo "$player_name - Score: $score | Correct: $correct_count | Incorrect: $incorrect_count" >>
"$SCORE_FILE"

    echo "Your score has been saved in $SCORE_FILE"

    break

fi

done

}

echo "Welcome to the Advanced Math Quiz Game :)"

echo "You have 10 seconds to answer each question."

echo "Let's begin...."

play_game
```

Screenshots of the output :

```
junayed_bin_karim@JUNAYED:~$ bash quiz.sh
Welcome to the Advanced Math Quiz Game :)
You have 10 seconds to answer each question.
Let's begin....
Enter your name: Junayed_Bin_Karim
What is 4 / 10? (Integer division)
Your answer:
Time's up
The correct answer was 0.
Score: 0 | Correct: 0 | Incorrect: 1
Level: Beginner
-----
Do you want to continue? (y/n): y
What is 9 + 10?
Your answer: 19
Correct!
Score: 1 | Correct: 1 | Incorrect: 1
Level: Beginner
-----
Do you want to continue? (y/n): y
What is 5 % 10? (Remainder)
Your answer: 0
Incorrect..... The correct answer was 5.
Score: 1 | Correct: 1 | Incorrect: 2
Level: Beginner
-----
Do you want to continue? (y/n): y
What is 7 - 10?
Your answer: -3
Correct!
Score: 2 | Correct: 2 | Incorrect: 2
Level: Beginner
-----
Do you want to continue? (y/n): y
What is 3 - 10?
Your answer: -7
Correct!
Score: 3 | Correct: 3 | Incorrect: 2
Level: Beginner
-----
Do you want to continue? (y/n): y
What is 5 squared?
Your answer: 25
Correct!
```

Score: 4 | Correct: 4 | Incorrect: 2
Level: Beginner

Do you want to continue? (y/n): y
What is 6 % 1? (Remainder)
Your answer: 1
Incorrect..... The correct answer was 0.
Score: 4 | Correct: 4 | Incorrect: 3
Level: Beginner

Do you want to continue? (y/n): y
What is 6 % 9? (Remainder)
Your answer: 3
Incorrect..... The correct answer was 6.
Score: 4 | Correct: 4 | Incorrect: 4
Level: Beginner

Do you want to continue? (y/n): y
What is 3 / 3? (Integer division)
Your answer: 0
Incorrect..... The correct answer was 1.
Score: 4 | Correct: 4 | Incorrect: 5
Level: Beginner

Do you want to continue? (y/n): y
Is 3 even or odd? (Type 'even' or 'odd')
Your answer: even
Incorrect! The correct answer was odd.
Score: 4 | Correct: 4 | Incorrect: 6
Level: Beginner

Do you want to continue? (y/n): y
What is 9 squared?
Your answer: 81
Correct!
You're improving.....
Score: 5 | Correct: 5 | Incorrect: 6
Level: Intermediate

Do you want to continue? (y/n): y
What is 15 - 9?
Your answer: 7
Incorrect..... The correct answer was 6.
Score: 5 | Correct: 5 | Incorrect: 7

Level: Intermediate

Do you want to continue? (y/n): y

What is $1 \% 8$? (Remainder)

Your answer: 1

Correct!

Score: 6 | Correct: 6 | Incorrect: 7

Level: Intermediate

Do you want to continue? (y/n): y

What is $9 * 20$?

Your answer: 180

Correct!

Score: 7 | Correct: 7 | Incorrect: 7

Level: Intermediate

Do you want to continue? (y/n): y

What is $5 \% 1$? (Remainder)

Your answer: 1

Incorrect..... The correct answer was 0.

Score: 7 | Correct: 7 | Incorrect: 8

Level: Intermediate

Do you want to continue? (y/n): y

What is $3 * 1$?

Your answer: 3

Correct!

Score: 8 | Correct: 8 | Incorrect: 8

Level: Intermediate

Do you want to continue? (y/n): y

What is $17 \% 11$? (Remainder)

Your answer: 5

Incorrect..... The correct answer was 6.

Score: 8 | Correct: 8 | Incorrect: 9

Level: Intermediate

Do you want to continue? (y/n): y

What is $3 \% 10$? (Remainder)

Your answer: 1

Incorrect..... The correct answer was 3.

Score: 8 | Correct: 8 | Incorrect: 10

Level: Intermediate

```
-----
Do you want to continue? (y/n): y
What is 3 % 10? (Remainder)
Your answer: 1
Incorrect..... The correct answer was 3.
Score: 8 | Correct: 8 | Incorrect: 10
Level: Intermediate
-----
Do you want to continue? (y/n): y
What is 11 + 12?
Your answer: 23
Correct!
Score: 9 | Correct: 9 | Incorrect: 10
Level: Intermediate
-----
Do you want to continue? (y/n): y
What is 17 % 2? (Remainder)
Your answer:
Time's up
The correct answer was 1.
Score: 9 | Correct: 9 | Incorrect: 11
Level: Intermediate
-----
Do you want to continue? (y/n): y
What is 15 + 20?
Your answer: 35
Correct!
You're doing really well.....:)
Score: 10 | Correct: 10 | Incorrect: 11
Level: Advanced
-----
Do you want to continue? (y/n): y
What is 42 - 6?
Your answer: 36
Correct!
Score: 11 | Correct: 11 | Incorrect: 11
Level: Advanced
-----
Do you want to continue? (y/n): n
Thanks for playing, Junayed_Bin_Karim!
Final Score: 11 | Correct: 11 | Incorrect: 11
Level: Advanced
Your score has been saved in math_quiz_scores.txt
junayed_bin_karim@JUNAYED:~$ |
```

This snippet highlights the use of conditional structures and user interaction.

3.2 Performance Analysis

- **Response Time:** The script efficiently generates questions and evaluates responses within the 10-second window, demonstrating minimal processing delay.
- **Scalability:** The modular approach ensures that additional question types or functionalities can be added with low overhead.
- **Resource Usage:** As a shell script, the project runs with minimal system resources, making it suitable for low-powered devices.

3.3 Results and Discussion

The game successfully challenges users with progressively difficult math problems. Key observations include:

- **User Engagement:** Immediate feedback and level rewards encourage continuous participation.
- **Accuracy:** The use of a timer effectively simulates a challenging quiz environment while ensuring the application gracefully handles timeouts.
- **Areas of Improvement:** Future enhancements can include more advanced mathematical operations, graphical interfaces, or integration with online leaderboards.

Chapter 4

Engineering Standards and Mapping

4.1 Impact on Society, Environment and Sustainability

4.1.1 Impact on Life

The project promotes educational engagement by providing an interactive tool that helps users practice arithmetic in a fun environment.

4.1.2 Impact on Society & Environment

By using open-source tools (Bash scripting) and minimal resources, the project encourages sustainable computing practices and digital literacy.

4.1.3 Ethical Aspects

The quiz game emphasizes fair play by giving equal chances for all users and providing clear feedback. There is an inherent respect for data privacy as user scores are stored locally.

4.1.4 Sustainability Plan

The lightweight nature of the script ensures that it can run on older hardware, reducing the need for frequent hardware upgrades and contributing to sustainable technology use.

4.2 Project Management and Team Work

The project involved clear task delegation including:

- Script development and debugging
- Designing user interaction flow
- Documentation and testing
Collaborative meetings and code reviews ensured adherence to quality standards and timely progress.

4.3 Complex Engineering Problem

Although the project is implemented in a scripting language, it models complex engineering problems by:

- Dynamically adjusting difficulty levels based on real-time performance.
- Integrating error handling (e.g., managing division by zero and timeouts).
- Structuring the code into distinct, reusable modules.

Chapter 5

Conclusion

5.1 Summary

The Advanced Math Quiz Game is an effective demonstration of Bash scripting capabilities in the context of an operating system lab project. It meets educational and functional objectives by providing a dynamic, interactive math quiz that adapts to the user's performance while teaching key programming concepts.

5.2 Limitation

- The game is limited to text-based interaction; a graphical interface could further enhance user engagement.
- The arithmetic operations are simple, and the game does not currently support decimal or complex calculations.
- The timer functionality, while effective, may sometimes interrupt users in cases of slow typing responses.

5.3 Future Work

Future enhancements could include:

- Incorporating advanced mathematical operations and a broader range of questions.
- Integrating a graphical user interface (GUI) for improved user experience.
- Extending the project to include IoT and cloud elements (aligning with CO1), such as remote monitoring of user scores or online leaderboards.
- Adding features like multiple levels or challenges that adjust more granularly based on performance.

References

1. Bash Scripting Guide – GNU Bash Manual
2. Unix Shell Programming – Advanced Bash-Scripting Guide
3. Time Management in Bash – [StackOverflow Discussions on read -t](#)