

CAPSTONE PROJECT FINAL REPORT

Dissertation submitted in fulfilment of the requirements for the Degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

-DATA SCIENCE WITH MACHINE LEARNING

BY

DHARA SAI GANESH

Registration No:12222126

Course code: CSE-228

Under the Guidance of Supervisor

Waseem Wani

School of computer science engineering

Lovely Professional University

Phagwara,Punjab(India) October

2023

DECLARATION STATEMENT

I hereby declare that the research work reported in the dissertation/dissertation proposal entitled **Quiz Generator** in partial fulfilment of the requirement for the award of Degree for Bachelor of Technology in Computer Science and Engineering at Lovely Professional University, Phagwara, Punjab is an authentic work carried out under supervision of my project supervisor **Mr. Waseem Ud Din Wani**.

I understand that the work presented herewith is in direct compliance with Lovely Professional University's Policy on plagiarism, intellectual property rights, and highest standards of moral and ethical conduct. Therefore, to the best of my knowledge, the content of this dissertation represents authentic and honest research effort conducted, in its entirety, by me. I am fully responsible for the contents of my dissertation work.

Signature of Candidate

Dhara Sai Ganesh

Reg.No: 12222126

SUPERVISOR'S CERTIFICATE

This is to certify that the work reported in the B. Tech Dissertation/dissertation proposal entitled, **Quiz Generator** submitted by **Dhara Sai Ganesh** at **Lovely Professional University, Phagwara, India** is a bonafide record of his original work carried out under my supervision. This work has not been submitted elsewhere for any other degree.

Signature of Supervisor

Lovely professional university,

School of computer science and Engineering,
Phagwara,Punjab

Date:

Acknowledgment:

I extend my heartfelt thanks to my professor, **Mr. Waseem Ud Din Wani** and for his invaluable support, guidance, and encouragement throughout this academic year.

Table of Contents

1. Abstract
2. Introduction
3. Objective of the project
4. System Overview
5. Description of the project
6. Implementation Details
7. Source Code
8. Input/Output
9. Scope of the project
10. Development
11. Conclusion

Abstract:

This java-based “Multi-Purpose Quiz Generator” project aims to simplify the process of quiz creation and management. By providing an intuitive platform For both quiz creators and participants, the project enables the generation of customized multiple-choice quizzes with efficient scoring and performance evaluation. Emphasizing user accessibility and seamless functionality the system’s adaptable design caters to diverse educational and assessment requirements, fostering interactive and engaging learning experiences.

Introduction

In an era where interactive learning is gaining precedence, the development of user-friendly educational tools remain paramount. The “Development of a Multi-Purpose Quiz Generator: A Java Implementation” represents a concreted effort to create an efficient and versatile platform for generating multiple-choice quizzes. This project aims to provide a seamless experience for both quiz creators and participants , facilitating then creation, management, and execution of quizzes in an intuitive manner.

Objectives

1. Develop a user- friendly quiz generator that allows users to create and manage, multiple-choice quizzes effortlessly.
2. Enable accurate scoring of user response to provide immediate feedback and assessment.
3. Implement a robust data management system to store and organise quiz questions, answers, and user responses securely.
4. Enhance the quiz-taking experience by providing clear instructions, intuitive interfaces, and informative feedback for participants.
5. Demonstrate the feasibility of the quiz generator through comprehensive testing and debugging procedures to ensure a seamless experience.

System Overview

The "Multi-Purpose Quiz Generator" is a Java-based application designed to facilitate the streamlined creation, management, and execution of multiple-choice quizzes. The system provides a user-friendly interface for quiz creators to input questions and corresponding answer options, enabling the generation of customized quizzes tailored to specific educational or assessment needs. Users can efficiently manage the quiz content, including adding, editing, and deleting questions as required.

For quiz takers, the system offers an interactive platform for executing the quizzes, presenting questions with multiple options and prompting users for their responses. The application incorporates an immediate feedback mechanism that assesses user answers, providing real-time evaluation and scoring based on the accuracy of the responses. Additionally, the system generates comprehensive performance feedback, including the total score and a performance category based on the achieved percentage, thereby fostering an engaging and informative quiz-taking experience.

The system architecture ensures robust data management, utilizing the Java Collections Framework to organize and store quiz questions, options, and user responses efficiently. Moreover, the application adheres to standard coding practices and incorporates error handling mechanisms to ensure stability and reliability throughout the quiz generation and execution processes. With its adaptable design and user-centric features, the "Multi-Purpose Quiz Generator" serves as a versatile tool for educators, students, and quiz enthusiasts, fostering an interactive and dynamic learning environment.

Description of the Project:

The "Multi-Purpose Quiz Generator" project is a Java-based application that serves as an all-encompassing tool for the seamless creation, management, and execution of multiple-choice quizzes. This user-centric system caters to the diverse needs of educators, students, and individuals seeking an

interactive platform for educational assessments, knowledge evaluations, and recreational quizzes.

Key features of the application include an intuitive quiz creation interface, allowing users to input customized questions and corresponding answer options. The system's robust data management capabilities enable efficient organization and storage of quiz data, ensuring easy access and modification of quiz

content as needed. Participants can engage with the quizzes through an interactive

and user-friendly interface, providing responses to the presented questions and receiving immediate feedback on their performance.

With an emphasis on user accessibility and streamlined functionality, the "Multi-Purpose Quiz Generator" fosters an engaging and dynamic learning environment. The project's architecture adheres to industry-standard coding practices, ensuring stability and reliability throughout the quiz generation and execution processes. By offering a comprehensive solution for quiz creation and assessment, the project aims to enhance the educational landscape by providing a versatile and adaptable tool for diverse learning needs and preferences.

Implementation Details:

1. Project Setup and Initialization:

- Set up the development environment with the required Java development tools and Integrated Development Environment (IDE) for coding.
- Create a new Java project to house the quiz generator application.

2. Define the Quiz Question Class:

- Define the Quiz Question class with private attributes for the question, options, and correct answer.
- Implement the constructor and getter methods to access the attributes.

3. Define the Quiz Class:

- Define the Quiz class with an ArrayList to store QuizQuestion objects.
- Implement methods for adding, removing, and managing quiz questions within the Quiz class.

4. Create User Interfaces:

- Design user interfaces for both quiz creators and participants, ensuring intuitive and user-friendly interactions.
- Implement input mechanisms for quiz creators to input questions, options, and correct answers.

5. Manage Data and Responses:

- Implement data management mechanisms using the Java Collections Framework to store and organize quiz questions and user responses.
- Develop algorithms to compare user responses with correct answers and calculate the quiz scores.

6. Integrate Error Handling and Exception Management:

- Implement error handling mechanisms to manage exceptions gracefully and ensure the stability of the application.
- Verify inputs, handle unexpected user behaviors, and provide informative error messages for a smooth user experience.

7. Test and Debug:

- Conduct comprehensive testing to validate the functionality and reliability of the quiz generator application.
- Debug any issues and refine the code to ensure optimal performance and accuracy.

Source Code:

Below is a simplified Java code snippet illustrating the lineage of a person in family tree.

```
import java.util.ArrayList;
import java.util.Scanner;

class QuizQuestion {
    private String question;
    private ArrayList<String> options;
    private int correctAnswer;

    public QuizQuestion(String question, ArrayList<String> options, int correctAnswer) {
        this.question = question;
        this.options = options;
        this.correctAnswer = correctAnswer;
    }

    public String getQuestion() {
        return question;
    }

    public ArrayList<String> getOptions() {
        return options;
    }

    public int getCorrectAnswer() {
        return correctAnswer;
    }
}

class Quiz {
    private ArrayList<QuizQuestion> questions;

    public Quiz() {
        this.questions = new ArrayList<>();
    }

    public void addQuestion(QuizQuestion question) {
        questions.add(question);
    }

    public String conductQuiz() {
        int score = 0;
        Scanner scanner = new Scanner(System.in);

        for (int i = 0; i < questions.size(); i++) {
            QuizQuestion currentQuestion = questions.get(i);

            System.out.println("Question " + (i + 1) + ": " +
currentQuestion.getQuestion());
```

```

        ArrayList<String> options = currentQuestion.getOptions();
        for (int j = 0; j < options.size(); j++) {
            System.out.println((j + 1) + ": " + options.get(j));
        }

        System.out.println();

        System.out.print("Enter your answer (1 or 2 or 3 or 4): ");
        int userAnswer = scanner.nextInt();

        if (userAnswer == currentQuestion.getCorrectAnswer()) {
            System.out.println("Correct!");
            score++;
        } else {
            System.out.println("Wrong! The correct answer is: " +
currentQuestion.getCorrectAnswer());
        }
    }

    int totalQuestions = questions.size();
    int percentageScore = (score * 100) / totalQuestions;

    String performanceFeedback;
    if (percentageScore >= 0 && percentageScore <= 40) {
        performanceFeedback = "Your performance is poor";
    } else if (percentageScore >= 41 && percentageScore <= 75) {
        performanceFeedback = "Your performance is intermediate";
    } else {
        performanceFeedback = "Your performance is excellent";
    }

    return "Your score is: " + score + "/" + totalQuestions + " (" +
percentageScore + "%). " + performanceFeedback;
}

}

public class Main {
    public static void main(String[] args) {
        Quiz quiz = new Quiz();

        ArrayList<String> options1 = new ArrayList<>();
        options1.add("Both are the same");
        options1.add("== compares object references while .equals() compares object
values");
        options1.add(".equals() compares object references while == compares object
values");
        options1.add(" Both compare object values");
        QuizQuestion question1 = new QuizQuestion(" What is the difference between ==
and .equals() in Java?", options1, 2);
        quiz.addQuestion(question1);

        ArrayList<String> options2 = new ArrayList<>();
        options2.add("45");
        options2.add("13");
        options2.add("25");
        options2.add("14");
        QuizQuestion question2 = new QuizQuestion("What will be the output of the
following code?"

```

```

+
        "" +
        "public class Test {\n" +
        "    public static void main(String[] args) {\n" +
        "        int x = 5;\n" +
        "        System.out.println(x++ + ++x);\n" +
        "    }\n" +
        "}", options2, 2);
quiz.addQuestion(question2);

ArrayList<String> options3 = new ArrayList<>();
options3.add("true");
options3.add("false");
options3.add("compilation error");
options3.add("none of these");
QuizQuestion question3 = new QuizQuestion("What will be the output of the
following code?" +
        "" +
        "public class Test {\n" +
        "    public static void main(String[] args) {\n" +
        "        String s1 = \"Hello\";\n" +
        "        String s2 = \"Hello\";\n" +
        "        System.out.println(s1 == s2);\n" +
        "    }\n" +
        "}", options3, 1);
quiz.addQuestion(question3);

ArrayList<String> options4 = new ArrayList<>();
options4.add("ArrayList is faster for insertion and deletion, while LinkedList
is faster for random");
options4.add("ArrayList is faster for random access, while LinkedList is faster
for insertion and deletion");
options4.add("Both have similar performance characteristics");
options4.add("ArrayList is a synchronized collection, while LinkedList is
not");
QuizQuestion question4 = new QuizQuestion("What is the difference between
ArrayList and LinkedList in Java?", options4, 2);
quiz.addQuestion(question4);

ArrayList<String> options5 = new ArrayList<>();
options5.add("Both are used to handle exceptions");
options5.add("throw is used to declare an exception, while throws is used to
throw an exception explicitly");
options5.add("throw is used to throw an exception explicitly, while throws is
used to declare an exception");
options5.add("Both are used interchangeably to throw exceptions");
QuizQuestion question5 = new QuizQuestion("What is the difference between throw
and throws in Java?", options5, 3);
quiz.addQuestion(question5);

String report = quiz.conductQuiz();

System.out.println("Quiz Report:\n");
System.out.println(report);
    }
}

```

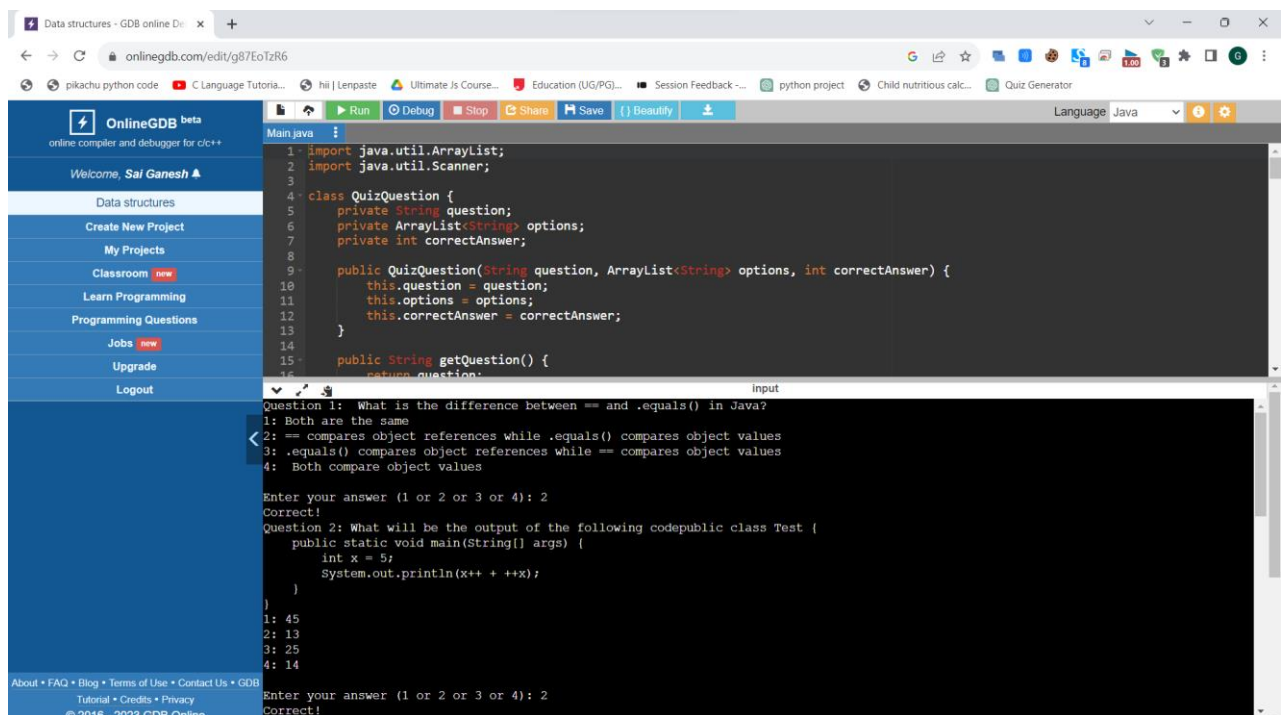
Input/Output:

Input Structure:

1.Quiz Creation Interface:

- * User inputs the quiz questions.
- * User inputs the multiple-choice options for each question.
- * User specifies the correct answer for each question.
- * User selects the answers for each question presented during the quiz.

OUTPUT:



```
1 import java.util.ArrayList;
2 import java.util.Scanner;
3
4 class QuizQuestion {
5     private String question;
6     private ArrayList<String> options;
7     private int correctAnswer;
8
9     public QuizQuestion(String question, ArrayList<String> options, int correctAnswer) {
10         this.question = question;
11         this.options = options;
12         this.correctAnswer = correctAnswer;
13     }
14
15     public String getQuestion() {
16         return question;
17     }
18 }
19
20 public class Test {
21     public static void main(String[] args) {
22         int x = 5;
23         System.out.println(x++ + ++x);
24     }
25 }
26
```

Question 1: What is the difference between == and .equals() in Java?

1: Both are the same
2: == compares object references while .equals() compares object values
3: .equals() compares object references while == compares object values
4: Both compare object values

Enter your answer (1 or 2 or 3 or 4): 2
Correct!

Question 2: What will be the output of the following code?

```
public class Test {
    public static void main(String[] args) {
        int x = 5;
        System.out.println(x++ + ++x);
    }
}
```

1: 45
2: 13
3: 25
4: 14

Enter your answer (1 or 2 or 3 or 4): 2
Correct!

OnlineGDB beta
online compiler and debugger for c/c++

Welcome, Sai Ganesh

Data structures

Create New Project

My Projects

Classroom new

Learn Programming

Programming Questions

Jobs new

Upgrade

Logout

About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy © 2016 - 2023 GDB Online

Main.java

```
1: import java.util.ArrayList;
2: import java.util.Scanner;
3:
4: class QuizQuestion {
5:     private String question;
6:     private ArrayList<String> options;
7:     private int correctAnswer;
8:
9:     public QuizQuestion(String question, ArrayList<String> options, int correctAnswer) {
10:         this.question = question;
11:         this.options = options;
12:         this.correctAnswer = correctAnswer;
13:     }
14:
15:     public String getQuestion() {
16:         return question;
17:     }
18: }
```

input

Question 3: What will be the output of the following code?

```
public class Test {
    public static void main(String[] args) {
        String s1 = "Hello";
        String s2 = "Hello";
        System.out.println(s1 == s2);
    }
}
```

1: true
2: false
3: compilation error
4: none of these

Enter your answer (1 or 2 or 3 or 4): 1
Correct!

Question 4: What is the difference between ArrayList and LinkedList in Java?

1: ArrayList is faster for insertion and deletion, while LinkedList is faster for random
2: ArrayList is faster for random access, while LinkedList is faster for insertion and deletion
3: Both have similar performance characteristics
4: ArrayList is a synchronized collection, while LinkedList is not

OnlineGDB beta
online compiler and debugger for c/c++

Welcome, Sai Ganesh

Data structures

Create New Project

My Projects

Classroom new

Learn Programming

Programming Questions

Jobs new

Upgrade

Logout

About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy © 2016 - 2023 GDB Online

Main.java

```
1: import java.util.ArrayList;
2: import java.util.Scanner;
3:
4: class QuizQuestion {
5:     private String question;
6:     private ArrayList<String> options;
7:     private int correctAnswer;
8:
9:     public QuizQuestion(String question, ArrayList<String> options, int correctAnswer) {
10:         this.question = question;
11:         this.options = options;
12:         this.correctAnswer = correctAnswer;
13:     }
14:
15:     public String getQuestion() {
16:         return question;
17:     }
18: }
```

Enter your answer (1 or 2 or 3 or 4): 3
Wrong! The correct answer is: 2

Question 5: What is the difference between throw and throws in Java?

1: Both are used to handle exceptions
2: throw is used to declare an exception, while throws is used to throw an exception explicitly
3: throw is used to throw an exception explicitly, while throws is used to declare an exception
4: Both are used interchangeably to throw exceptions

Enter your answer (1 or 2 or 3 or 4): 3
Correct!

Quiz Report:

Your score is: 4/5 (80%). Your performance is excellent

...Program finished with exit code 0
Press ENTER to exit console.

Scope

The scope of this project is to provide a basic Quiz generation system. It covers essential features for generating the questions , edit options in the interface , see the score and as well as percentage of the score. Also feedback is shared with user according to their scores.

Development

The development section outlines the tools and technologies used to create the project. It discusses the software development environment, libraries, and the step by-step process of developing the quiz generating system. Here's a step-by-step breakdown of the code's development:

- 1.Project Initiation
- 2.Environment Setup
- 3.Project Structure
- 4.Coding the Appointment Class
- 5.Coding the Main Class
- 6.User Interface Design
- 7.Input Handling
- 8.Answer the questions from the options
9. Viewing scores and feedback of the score
- 10.Exiting the Program
- 11.Invalid Input Handling
- 12.Testing and Debugging

13.Documentation and Comments

14.Finalization

15.Version Control

16.Deployment

Conclusion

The "Multi-Purpose Quiz Generator" project has successfully addressed the need for a user-friendly and efficient platform for the creation, management, and execution of multiple-choice quizzes. Through the implementation of Java-based data management and interactive user interfaces, the project has demonstrated its capability to streamline the process of quiz generation and assessment, catering to the diverse requirements of educators, students, and quiz enthusiasts.

The project's emphasis on intuitive user interfaces, robust data handling, and immediate feedback mechanisms has contributed to the creation of an engaging and dynamic learning environment. By providing a comprehensive solution for both quiz creators and participants, the "Multi-Purpose Quiz Generator" has showcased its potential in fostering interactive educational experiences and facilitating effective knowledge assessments.

Moving forward, the project's adaptable design and scalable architecture lay the foundation for potential enhancements, including the integration of advanced features, user authentication, and result tracking. The "Multi-Purpose Quiz Generator" stands as a testament to the significance of interactive educational tools in promoting a collaborative and engaging learning ecosystem, paving the way for future innovations in the field of educational technology.