**Programming Assignment Unit 1**

University of the People

CS 1102-01: Programming 1 AY2024-T4

Romana Riyaz

22 November 2024

# Programming Assignment Unit 1

## Problem Statement

Create a Java program that simulates a simple quiz game. The program should prompt the user with a series of questions and allow them to enter their answers. After the user has answered all the questions, the program should compute and display the final score based on the number of correct answers.

Write a Java program that implements the functionality described in the scenario. Ensure that the program is error-free, compiles successfully, and produces the expected output. Test the program with different inputs to verify its correctness.

The following requirements must be met:

* The program should include five multiple-choice questions with four options.
* The four options should be labeled A, B, C, and D.
* The program should prompt the user to enter their answer by typing the corresponding letter (A, B, C, or D).
* After the user has entered their answer for each question, the program should compare it to the correct answer and keep track of the number of correct responses.
* The program should compute and display the final score as a percentage of the total number of questions.
* Use if and switch case statements to handle user input and compare it to the correct answers.
* Include comments to explain the purpose of each section of code and enhance code readability.

**Abstract**

The "QuizGame" Java program is a console-based quiz game that tests the user's knowledge on various topics. It prompts the user with a series of multiple-choice questions, presents options for each question, and records the user's responses. The program then evaluates the user's answers against a set of predefined correct answers and calculates a final score based on the number of correct responses. Utilizing the Scanner class for user input and arrays to store questions, options, and correct answers, the QuizGame provides an interactive and engaging way to challenge and assess the user's knowledge.

**Solution Code (VSCode Snapshot)**

**Solution Code (VSCode Text)**

// import the Scanner class

import java.util.Scanner;

public class QuizGame {

public static void main(String[] args) {

// Create a Scanner object to take user input

Scanner scanner = new Scanner(System.in);

int score = 0; // Initialize the score counter to 0

// Define the questions, options, and correct answers

String[] questions = {

"Which data structure follows LIFO?", // Q1

"Which sorting algorithm has O(n log n) average-case time complexity?", // Q2

"Which language is primarily used for statistical computing?", // Q3

"What language is used to create web pages?", // Q4

"Which protocol retrieves email from a server?" // Q5

};

String[][] options = {

// Options for Q1

{"A) Queue", "B) Stack", "C) Linked List", "D) Tree"},

// Options for Q2

{"A) Bubble Sort", "B) Insertion Sort", "C) Merge Sort", "D) Selection Sort"},

// Options for Q3

{"A) Java", "B) Python", "C) R", "D) C++"},

// Options for Q4

{"A) HTML", "B) HTML", "C) HTTP", "D) HTTPS"},

// Options for Q5

{"A) HTTP", "B) FTP", "C) SMTP", "D) POP3"}

};

// Array containing correct answers for each question

char[] correctAnswers = {'B', 'C', 'C', 'A', 'D'}; // Correct answers in uppercase

// Display and process each question

for (int i = 0; i < questions.length; i++) {

// Print the question

System.out.println(questions[i]);

for (String option : options[i]) {

// Print the options for the current question

System.out.println(option);

}

// Prompt the user for input

System.out.print("Enter your answer (A, B, C, or D): ");

// Read user's answer and convert to uppercase

char userAnswer = Character.toUpperCase(scanner.next().charAt(0));

// Compare user's answer with correct answer and update score

if (userAnswer == correctAnswers[i]) {

score++; // Increment score if the answer is correct

}

}

// Calculate and display the final score

// Calculate percentage score

double percentageScore = (double) score / questions.length \* 100;

// Print the final score

System.out.println("Your final score: " + percentageScore + "%");

// Close the scanner object to release resources

scanner.close();

}

}

**Code Explanation**

**1. Import Statement:**

// import the Scanner class

import java.util.Scanner;

This line imports the Scanner class from the java.util package. The Scanner class is used to obtain input from the user.

**2. Main Method:**

public static void main(String[] args) {

// Code block

}

This is the main method, where the execution of the program begins. It contains the primary logic for the quiz game.

**3. Scanner Initialization:**

Scanner scanner = new Scanner(System.in);

This line initializes a Scanner object named scanner to read input from the standard input stream (usually the keyboard).

**4. Arrays for Questions, Options, and Correct Answers:**

String[] questions = {..};

String[][] options = {..};

char[] correctAnswers = {..};

These arrays store the questions, options, and correct answers for the quiz game. Each element of the questions array corresponds to a question, and the elements of the options array correspond to the multiple-choice options for each question. The correctAnswers array stores the correct answer for each question.

**5. Display and Process Questions:**

for (int i = 0; i < questions.length; i++) {

// Code block

}

This for loop iterates through each question in the questions array. Inside the loop, each question is displayed along with its options, and the user is prompted to enter their answer.

**6. Comparison with Correct Answers:**

if (userAnswer == correctAnswers[i]) {

score++;

}

This if statement compares the user's answer (userAnswer) with the correct answer for the current question. If the user's answer matches the correct answer, the score is incremented.

**7. Calculation of Final Score:**

double percentageScore = (double) score / questions.length \* 100;

After processing all questions, this line calculates the user's final score as a percentage of correct answers.

**8. Displaying the Final Score:**

System.out.println("Your final score: " + percentageScore + "%");

Finally, the user's final score is displayed on the console.

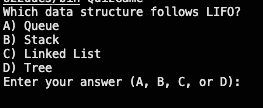
**9. Closing Scanner:**

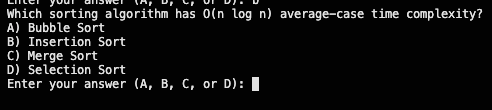
scanner.close();

This line closes the Scanner object to release system resources associated with it, preventing resource leaks.

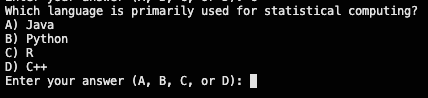
**Summary**

The program implements a simple quiz game that engages users with multiple-choice questions, evaluates their responses, and computes a final score. Key components of the code include the use of the Scanner class to capture user input, arrays to organize questions, options, and correct answers, and control structures such as loops and conditionals to manage the quiz flow. Through clear organization and logical structuring, the program offers an interactive and educational experience for users to test their knowledge on various topics. Additionally, by closing the Scanner object at the end of execution, the code ensures efficient resource management.

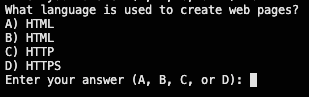
**Output of the Code**



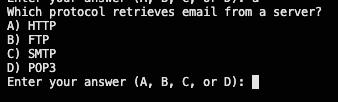
*Question 1 output*

**

*Question 2 output*

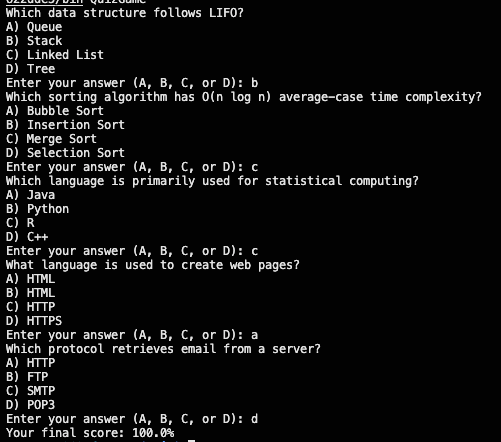


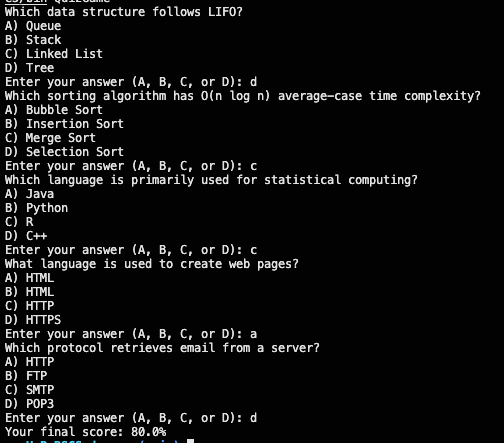
*Question 3 output*

**

*Question 4 output*

*Question 5 output*

**Output scored with correct answers**

****

**Output scored with in-correct answers**

**References**

Oracle Java Documentation. *Official documentation for the Scanner class.* Scanner Class

*https://docs.oracle.com/javase/8/docs/api/java/util/Scanner.html*

Oracle Java Tutorials. *A Tutorial on arrays in Java.* Arrays

*https://docs.oracle.com/javase/tutorial/java/nutsandbolts/arrays.html*

Oracle Java Tutorials. *Guide on control flow statements.* Control Flow Statements

*https://docs.oracle.com/javase/tutorial/java/nutsandbolts/flow.html*

GeeksforGeeks. *Guide on Java Arrays.* Java Arrays

*https://www.geeksforgeeks.org/arrays-in-java/*

GeeksforGeeks. *Control Flow Statements.* Java Control Statements

*https://www.geeksforgeeks.org/java-control-statements/*