

# **Online Quiz Application Using Java**

**PRESENTED BY**

**CHALLA SWATHI**

**Bachelor of Technology (Computer  
Science Engineering),**

**G Pulla Reddy Engineering College,  
Kurnool Andhra Pradesh.**

# CONTENTS

ABSTRACT .....	i
CHAPTER 1 .....	1
1.1 INTRODUCTION .....	1
1.2 Problem Statement .....	1
1.3 System Requirements.....	1
CHAPTER 2 .....	2
2.1 IMPLEMENTATION OF CODE USING IntelliJ IDEA .....	2
2.2 CODE EXPLANATION .....	2
2.3 Source Code .....	3
2.3.1 Online Quiz App .....	3
2.3.2 Question Class .....	6
2.3.3 Quiz Class.....	7
2.3.4 Student Class .....	10
REFERENCES .....	11

# **ABSTRACT**

The online Quiz application is a simple console-based Quiz System. Users will be presented with a series of questions and each question has 4 different options and displaying the total score after completion of Quiz Successfully. The Project is Implemented with Basic Concepts of Java And Collections like Array List etc. The project is evaluated with the user responses and provides the feedback based on their scores.

# CHAPTER 1

## 1.1 INTRODUCTION

The primary objective of this project is to develop a Quiz application using Java. The Project IntelliJ IDEA is an integrated development environment (IDE) written in Java for developing computer software written in Java, Kotlin, Groovy, and other JVM-based languages.

The Project Contains a Question class which containing attributes such as question text, options which are stored in Array List and the correct answer. The Quiz Class contains List of questions stored in Array List as Array of objects. The Student Class which contains Name and roll number of the student. The main Method which is OnlineQuizApp class evaluates the total score based on user answer and displaying the total score After successfully attempted the test. The Project ensure the proper validation of user input to handle potential errors gracefully.

## 1.2 Problem Statement

The objective of Online Quiz application is to facilitate user friendly platform by reducing the manual effort. In past days we have seen that quiz were taken manually but with rise of technology we are able to conduct them and generate scores automatically. Especially in the time of this pandemic use of such applications is highly appreciated as schools and colleges are very much affected.

## 1.3 System Requirements

1. Windows or Linux
2. JDK
3. IntelliJ IDEA

# CHAPTER 2

## 2.1 IMPLEMENTATION OF CODE USING IntelliJ IDEA

Download the Latest version of IntelliJ IDEA from <https://www.jetbrains.com/idea/> website. After downloading Click on new Project and then give name for the Project and then click on create to create the Project. Then create the Question, Quiz, OnlineQuizApp and Student classes and implement the logic as per the Project Requirements.

## 2.2 CODE EXPLANATION

The Quiz Class contains mainly 10 questions and each questions have 4 options. The Quiz class contains Questions which are structured using Questions class with the help of constructor and Stored each question in Array List which is useful for the retrieving the questions in OnlineQuizApp. By Taking the Input from the console create the student object. The code is validated for every input taking from the student input using regex expressions patterns. Suppose if the student enters the invalid name and roll number the console display the message of entering the wrong details and again student has to start quiz again. After taking input from the console each question is displayed with the list of options. Then the student must choose the one option and when student choose the option, the console displays whether the student choose correct answer or not by displaying some message in console and then next question is displayed vice versa and finally displays the total score and feedback after successful completion of quiz.

## 2.3 Source Code

### 2.3.1 Online Quiz App

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class OnlineQuizApp {
    public static boolean flag = false;

    //checks for name contains only alphabets
    private static boolean validateAlpha(String str) {
        return ((str != null)
            && (!str.equals(""))
            && (str.matches("[a-zA-Z]*$")));
    }

    //checks for rollno contains alpha_numeric charcaters
    private static boolean validateAlphaNumeric(String str) {
        Pattern pattern = Pattern.compile("[a-zA-Z0-9]*");
        Matcher matcher = pattern.matcher(str);

        return ((str != null)
            && (!str.equals(""))
            && matcher.matches());
    }

    //checks whether entered option conatins only digits and enter between 0
    and 4
    private static boolean validateDigits(String str) {
        return ((str.length() == 1)
            && (str != null)
            && (!str.equals(""))
            && (str.equals("1") || str.equals("2") || str.equals("3")
            || str.equals("4")))
        );
    }

    private static void quizInvoke() throws IOException {

        BufferedReader br = new BufferedReader(new
        InputStreamReader(System.in));
        char ch = 'y';

        do{
            // count of the student score
            int score = 0;

            System.out.println("Please enter your name: (should contain
            only alphabets)");

            String studentName = br.readLine();
            if(!validateAlpha(studentName)) {
                System.out.println("Enter the valid name containing only
            alphabets");
                flag = true;
                break;
            }
        }
    }
}
```

```

    }

    System.out.println("Please enter your roll number: ");

    String studentRollNumber = br.readLine();
    if(!validateAlphaNumeric(studentRollNumber)) {
        System.out.println("Enter the valid roll number
containing only alpha-numerics");
        flag = true;
        break;
    }
    Student student = new Student(studentName,
studentRollNumber);
    System.out.println("Please answer the following questions:
");

    Quiz studentQuiz = new Quiz();
    ArrayList<Question> studentQuestions =
studentQuiz.getQuestions();

    for(int i=0; i<studentQuestions.size(); i++) {
        System.out.print("Question " + (i+1) + ": ");

System.out.println(studentQuestions.get(i).getQuestionText());

        ArrayList<String> studentQuizOptions =
studentQuestions.get(i).getOptions();
        for(int j=0; j<studentQuizOptions.size(); j++) {
            System.out.println((j+1) + ": " +
studentQuizOptions.get(j));
        }
        System.out.println("Please enter your option (in numbers
1,2,3,4): ");

        String str_enteredOption = br.readLine();

        if(!validateDigits(str_enteredOption)) {
            System.out.println("Enter the valid option between 1
and 4 (both are inclusive)");
            flag = true;
            break;
        }
        int enteredOption = Integer.parseInt(str_enteredOption);

        // for finding the crct answers's index
        int crctOption =
studentQuizOptions.indexOf(studentQuestions.get(i).getCorrectAnswer());

        // checking the enteredOption with crctOption
        if(enteredOption == crctOption+1) {
            score++;
            System.out.println("Great! It is the correct
answer!!");
        } else {
            System.out.println("Yikes.. its the wrong answer!");
        }
    }
    if(flag) {
        break;
    }

```

```

    }

    System.out.println("Student: " + student.getName() + " with
Roll number: " + student.getRollNumber() + " score is " + score);
    if(score>=5 && score<=8)
    {
        System.out.println("Your score is Good");
    }
    else if(score<5)
    {
        System.out.println("Your score is to low");
    }
    else
    {
        System.out.println("Excellent Peformance");
    }

    System.out.print("\n\n-----\n\n");

    System.out.println("Do you want to take the quiz again? (enter 'y' or any
other key)");

    ch = (char)br.read();
    // error detection to get rid of default java behavoiur
    br.readLine();
    // OR else dont use int input instead take input as string
and convert it into int
    // then the dummy readline is not required
    // ex:- ch = br.readLine().charAt(0);

    if(ch!='y' && ch!='Y') {
        br.close();
    }
    }while(ch=='y' || ch=='Y');

    if(!flag) {
        System.out.println("Quiz attempted successfully!");
    }
}

public static void main(String[] args) throws IOException {
    do {
        flag = false;
        quizInvoke();
    }while(flag);
}
}

```



### 2.3.2 Question Class

```
import java.util.ArrayList;

public class Question {
    private String questionText;
    private ArrayList<String> options;
    private String correctAnswer;

    public Question(String questionText, ArrayList<String> options,
String correctAnswer) {
        this.questionText = questionText;
        this.options = options;
        this.correctAnswer = correctAnswer;
    }

    public String getQuestionText() {
        return questionText;
    }

    public void setQuestionText(String questionText) {
        this.questionText = questionText;
    }

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

    public void setOptions(ArrayList<String> options) {
        this.options = options;
    }

    public String getCorrectAnswer() {
        return correctAnswer;
    }

    public void setCorrectAnswer(String correctAnswer) {
        this.correctAnswer = correctAnswer;
    }
}
```

### 2.3.3 Quiz Class

```
import java.util.ArrayList;

public class Quiz {

    private ArrayList<Question> questions = new ArrayList<>();

    public Quiz() {
        String questionText1 = "Who invented Java? ";
        ArrayList<String> options1 = new ArrayList<>();
        options1.add("Guido van Rossum");
        options1.add("James Gosling");
        options1.add("Dennis Ritchie");
        options1.add("Bjarne Stroustrup");
        String correctAnswer1 = "James Gosling";
        Question question1 = new Question(questionText1, options1,
correctAnswer1);
        this.questions.add(question1);

        String questionText2 = "Which of the following is not a keyword?
";
        ArrayList<String> options2 = new ArrayList<>();
        options2.add("this");
        options2.add("continue");
        options2.add("class");
        options2.add("it");
        String correctAnswer2 = "it";
        Question question2 = new Question(questionText2, options2,
correctAnswer2);
        this.questions.add(question2);

        String questionText3= "Identify the corrected definition of a
package? ";
        ArrayList<String> options3 = new ArrayList<>();
        options3.add("A package is a collection of editing tools");
        options3.add("A package is a collection of classes");
        options3.add("A package is a collection of classes and
interfaces");
        options3.add("A package is a collection of interfaces");
        String correctAnswer3 = "A package is a collection of classes and
interfaces";
        Question question3 = new Question(questionText3, options3,
correctAnswer3);
        this.questions.add(question3);

        String questionText4= "When is the object created with new
keyword? ";
        ArrayList<String> options4 = new ArrayList<>();
        options4.add("At run time");
        options4.add("At compile time");
        options4.add("Depends on the code");
        options4.add("None");
        String correctAnswer4 = " At run time ";
        Question question4 = new Question(questionText4, options4,
correctAnswer4);
```

```

        this.questions.add(question4);

        String questionText5 = "Identify the modifier which cannot be
used for constructor? ";
        ArrayList<String> options5 = new ArrayList<>();
        options5.add("Public");
        options5.add("Protected");
        options5.add("Private");
        options5.add("Static");
        String correctAnswer5 = "Static";
        Question question5 = new Question(questionText5, options5,
correctAnswer5);
        this.questions.add(question5);

        String questionText6 = "Which of the following is not a keyword?
";
        ArrayList<String> options6 = new ArrayList<>();
        options6.add("this");
        options6.add("continue");
        options6.add("class");
        options6.add("it");
        String correctAnswer6 = "it";
        Question question6 = new Question(questionText6, options6,
correctAnswer6);
        this.questions.add(question6);

        String questionText7= "Exception created by try block is caught
in which block? ";
        ArrayList<String> options7 = new ArrayList<>();
        options7.add("catch");
        options7.add("throw");
        options7.add("final");
        options7.add("none");
        String correctAnswer7 = "catch";
        Question question7 = new Question(questionText7, options7,
correctAnswer7);
        this.questions.add(question7);

        String questionText8= "Which of the following exception is thrown
when divided by zero statement is executed? ";
        ArrayList<String> options8 = new ArrayList<>();
        options8.add("NullPointerException");
        options8.add("NumberFormatException");
        options8.add("ArithmeticException");
        options8.add("None");
        String correctAnswer8 = "ArithmeticException";
        Question question8 = new Question(questionText8, options8,
correctAnswer8);
        this.questions.add(question8);

        String questionText9= "Which of the following is a type of
polymorphism in Java Programming? ";
        ArrayList<String> options9 = new ArrayList<>();
        options9.add("Multiple polymorphism");
        options9.add("Compile time polymorphism");
        options9.add("Multilevel polymorphism");
        options9.add("Execution time polymorphism");

```

```

        String correctAnswer9 = "Compile time polymorphism";
        Question question9 = new Question(questionText9, options9,
correctAnswer9);
        this.questions.add(question9);

        String questionText10= "What does the expression float a = 35 / 0
return?";
        ArrayList<String> options10 = new ArrayList<>();
        options10.add("0");
        options10.add("Not a Number");
        options10.add("infinity");
        options10.add("Run time Exception");
        String correctAnswer10 = "Run time Exception";
        Question question10 = new Question(questionText10, options10,
correctAnswer10);
        this.questions.add(question10);

    }

    public ArrayList<Question> getQuestions() {
        return questions;
    }

    public void setQuestions(ArrayList<Question> questions) {
        this.questions = questions;
    }
}

```

### 2.3.4 Student Class

```
public class Student {  
  
    private String name;  
    private String rollNumber;  
  
    public Student(String name, String rollNumber) {  
        this.name = name;  
        this.rollNumber = rollNumber;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
  
    public String getRollNumber() {  
        return rollNumber;  
    }  
  
    public void setRollNumber(String rollNumber) {  
        this.rollNumber = rollNumber;  
    }  
}
```

## REFERENCES

<https://www.geeksforgeeks.org/arraylist-in-java/>

<https://www.geeksforgeeks.org/java-io-bufferedreader-class-java/>

<https://www.programiz.com/java-programming/bufferedReader>

<https://www.javamadesoeasy.com/2015/12/how-to-check-string-is-alphanumeric-or.html>

<https://prutor.ai/check-if-a-string-contains-only-alphabets-in-java-using-regex/>