QUIZ BOWL app

Object oriented programming group assignment

Group members :

1. Yousef L.T Aldawoud (SES150025)
2. Ashraf Saleem (SES160001)
3. Anusheri Odiboev @ Bilal (SES160021)

Introduction

- Objective of the assignment is to make quiz bowl application that utilizes multiple class and inheritance in a practical manner.
- We made an extra application which is a question creator application

Quiz Bowl Application

- The quiz bowl application was developed as a quiz
- The questions imported from a file (txt file only).
- The user would than answer the questions based on the type given
- The question comes in form of MCQ, Short written answer and True and false.
- Questions can be skipped by typing in SKIP
- Results are shown at the end of the quiz.

Question Creating Application

- This application was developed to create questions and save them into a file (txt file only).
- The application requires a to be selected file to proceed with further steps at the start.
- You select the types of questions you want
- Fill in the relevant information
- File that’s been created can be used by quiz bowl application

Quiz Maker

What’s Quiz maker

- A program to make question and add it to a text file

Parts of the Quiz Maker code

- The main process
- GUI (Graphical user interface).
- Helping methods

The Main process

```
public void actionPerformed(ActionEvent arg0) {
    try {
        FileWriter writer = new FileWriter(fileLocation);

        int point=Integer.parseInt(points.getText());
        if(QuizBowl.checkDuplicate(allQuestions, Question.getText())){
        if(Types.getSelectedIndex()==0){
            if(MCC.getText().split("\n").length<9||MCC.getText().split("\n").length>3){


                if(contains(MCC.getText().split("\n"),correct.getText())){

                    allQuestions.add(new QuestionMC(MCC.getText().split("\n"),Question.getText(),point,correct.getText()));
                    result.setText("<html><h3 style='color:Blue'>Question was added succussfully");
                    //System.out.println(allQuestions.get(allQuestions.size()-1));

                }else{
                    result.setText("<html><h3 style='color:red'>The correct answer has to be one of the choices"
                            + "</h3></html>");
                }
            }
```

The Main process

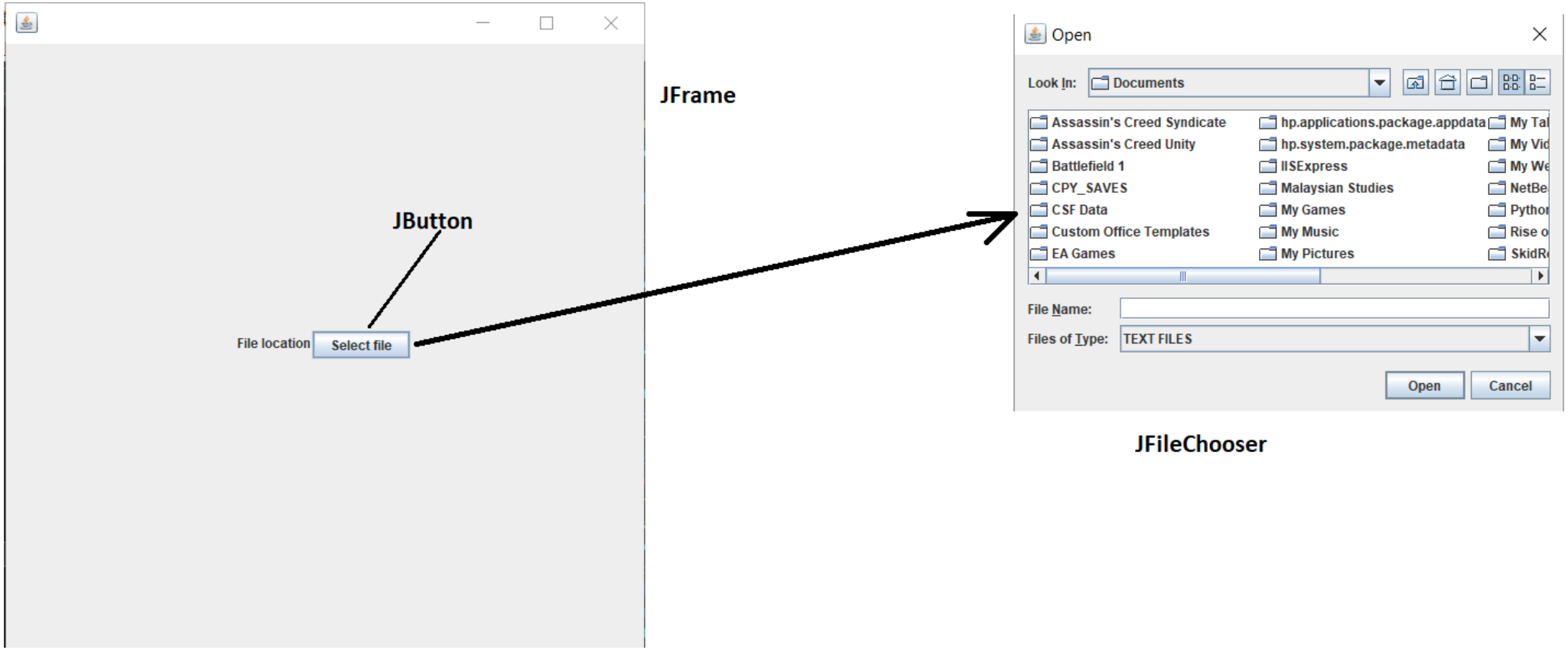
```
for(int i=0;i<allQuestions.size();i++){
    String typo=allQuestions.get(i).getClass().getName();
    if(typo.equals("QuestionTF")){
        QuestionTF q=((QuestionTF)allQuestions.get(i));
        c=c+q.insertForm()+"\r\n";
    }else if(typo.equals("QuestionSA")){
        QuestionSA q=((QuestionSA)allQuestions.get(i));
        c=c+q.insertForm()+"\r\n";
    }else if(typo.equals("QuestionMC")){
        QuestionMC q=((QuestionMC)allQuestions.get(i));
        c=c+q.insertForm()+"\r\n";
    }

}
writer.write(c);
writer.flush();
writer.close();
```

GUI –Main page

```java
public class AddingQuestions extends JPanel{
    private File file;
    private JFileChooser fileChooser =new JFileChooser();
    private JButton chooseFile=new JButton("Select file");
    protected String fileLocation;

    private String fileLoc;
    private JLabel pointsLabel=new JLabel("points");

    private JTextField points=new JTextField(30);
    private String [] types= {"MCQ","True/False","One word questions"};
    private String [] tf= {"True","False"};
    private JComboBox Types=new JComboBox(types);
    JLabel QuestionLabel=new JLabel("The question : ");
    JTextArea MCC=new JTextArea(2,9);
    JLabel MCCLabel=new JLabel("Choices");
    private JLabel fileLabel=new JLabel("File location");
    private JTextField Question=new JTextField(30);
    private JLabel correctLabel=new JLabel("Correct answer");
    private JTextField correct=new JTextField(30);
    private JComboBox trueOrFalse=new JComboBox(tf);
    private JButton Add=new JButton("Add a Question");
    private JLabel result=new JLabel();
    private String fileContent;
    private JButton submit=new JButton("Submit");
    private LinkedList<Object> allQuestions=new LinkedList<Object>();
    public AddingQuestions(){
        //making gui
```

```java
public AddingQuestions(){
    //making gui
    setLayout(new GridBagLayout());
    fileChooser.setFileFilter( new FileNameExtensionFilter("TEXT FILES", "txt", "text"));

    MCC.setVisible(false);
    Types.setVisible(false);
    QuestionLabel.setVisible(false);
    Question.setVisible(false);
    correct.setVisible(false);
    correctLabel.setVisible(false);
    pointsLabel.setVisible(false);
    points.setVisible(false);
    MCC.setVisible(false);
    Add.setVisible(false);
    MCCLabel.setVisible(false);
    GridBagConstraints xy=new GridBagConstraints();
    xy.fill=2;

    add(fileLabel,xy);
    xy.gridx=1;
    xy.gridy=0;
    xy.insets=new Insets(5,1,1,1);
    add(Types,xy);
    add(chooseFile,xy);
    xy.gridx=0;
    xy.gridy=1;
    add(QuestionLabel,xy);
    add(submit,xy);
    xy.gridx=1;
    xy.gridy=1;
    add(Question,xy);
    // true or false
```

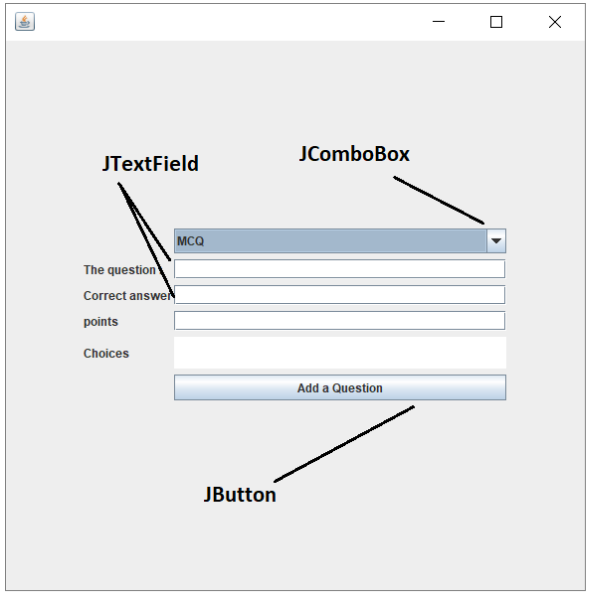
GUI –Main page

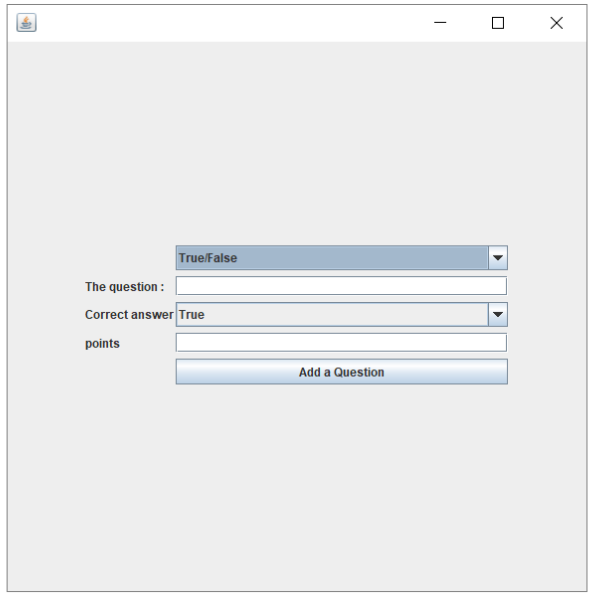


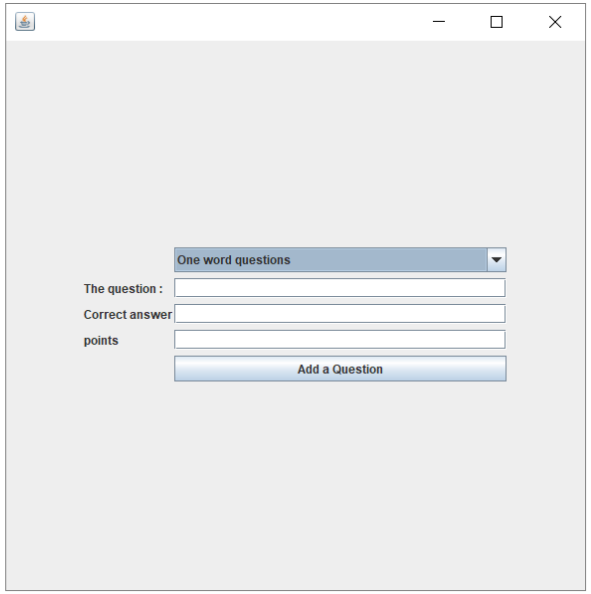
GUI –Question maker page

```
Types.addActionListener(new ActionListener (){

    @Override
    public void actionPerformed(ActionEvent e) {

        if(Types.getSelectedIndex()==0){
            trueOrFalse.setVisible(false);
            MCC.setVisible(true);
            MCCLabel.setVisible(true);
            correct.setVisible(true);
            result.setText("");
        }else if(Types.getSelectedIndex()==1){
            MCC.setVisible(false);
            trueOrFalse.setVisible(true);
            MCCLabel.setVisible(false);
            correct.setVisible(false);
            result.setText("");
        }else if(Types.getSelectedIndex()==2){
            trueOrFalse.setVisible(false);
            MCC.setVisible(false);
            MCCLabel.setVisible(false);
            correct.setVisible(true);
            result.setText("");
        }
    }
```

GUI –Question maker page



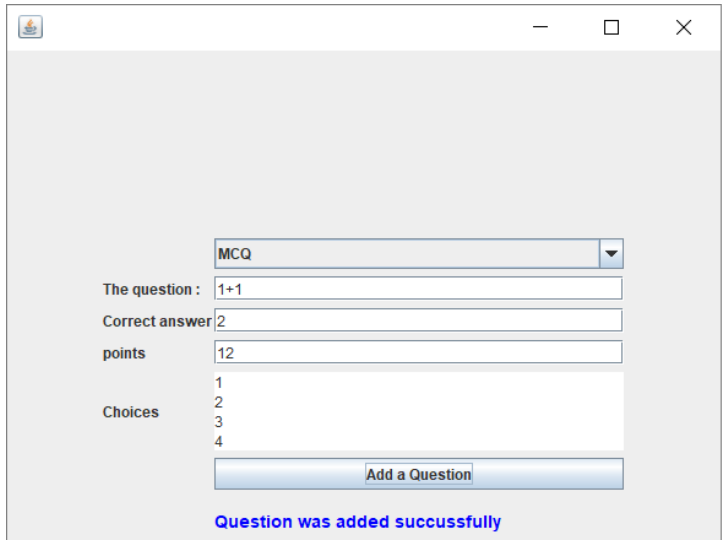




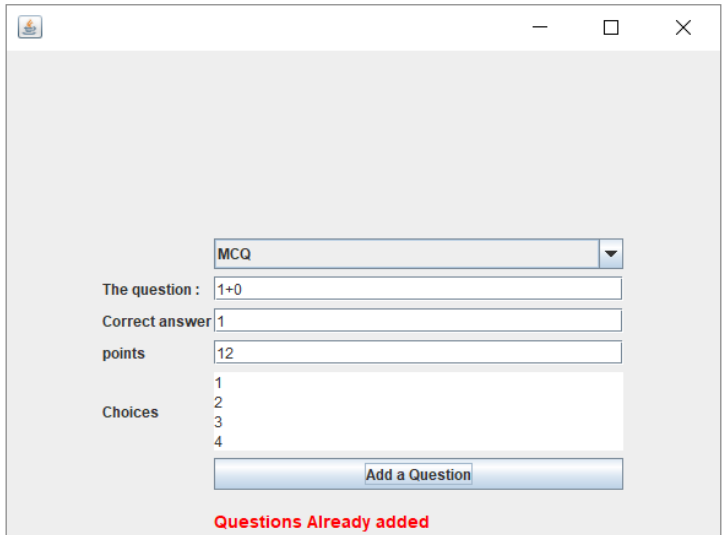
Helping methods

```
public boolean checkQuestion(String question,String fileLoc){
    for(int i=0;allQuestions.size()>i;i++){
        if(((Question)allQuestions.get(i)).getQues().equals(question)){
            return false;
        }
    }
    return true;
}
public static boolean contains(String[] abcde, String e) {
    for(String x:abcde){
        if(x.equals(e)){
            return true;
        }
    }
    return false;
}
```

Helping methods

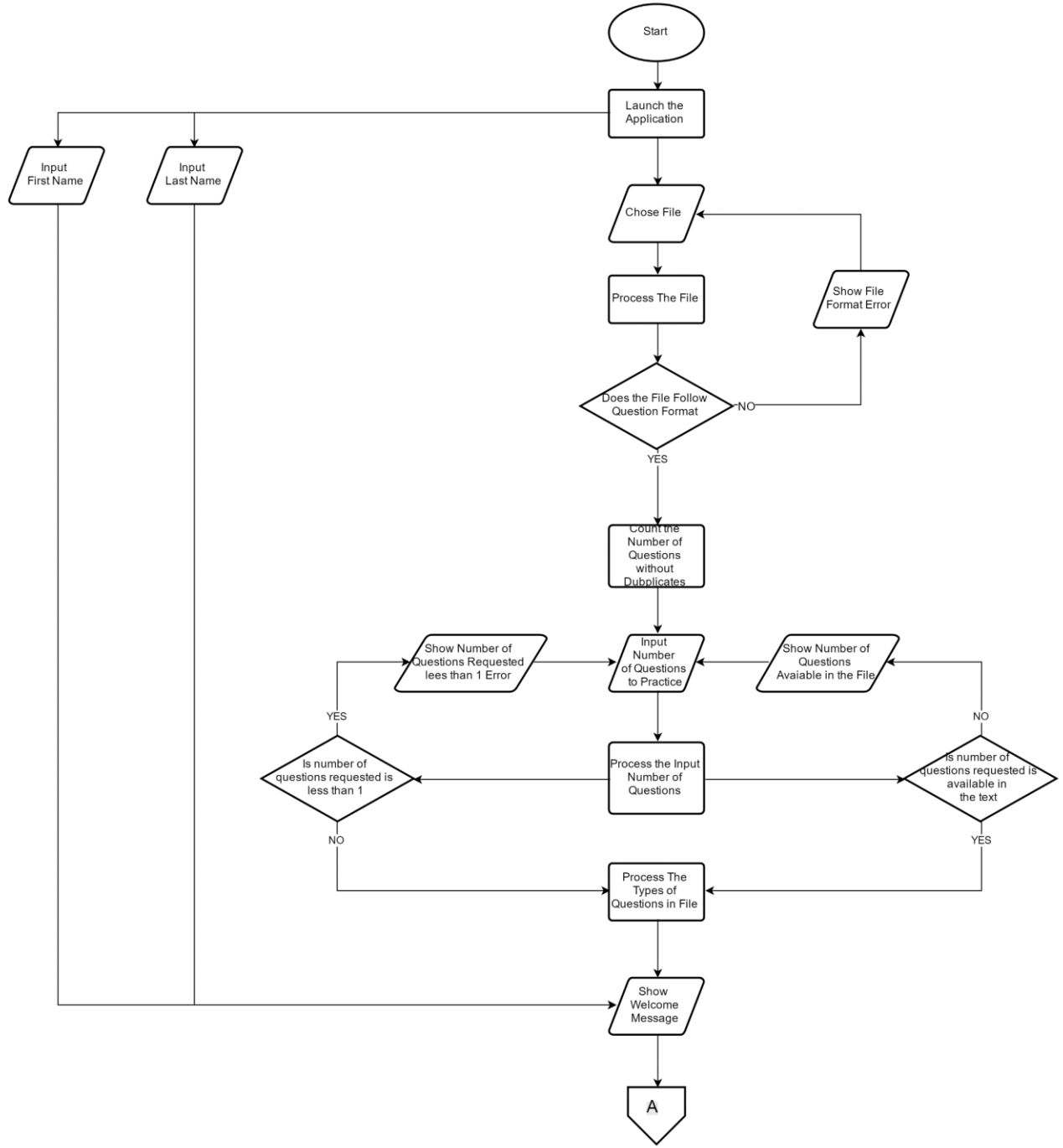


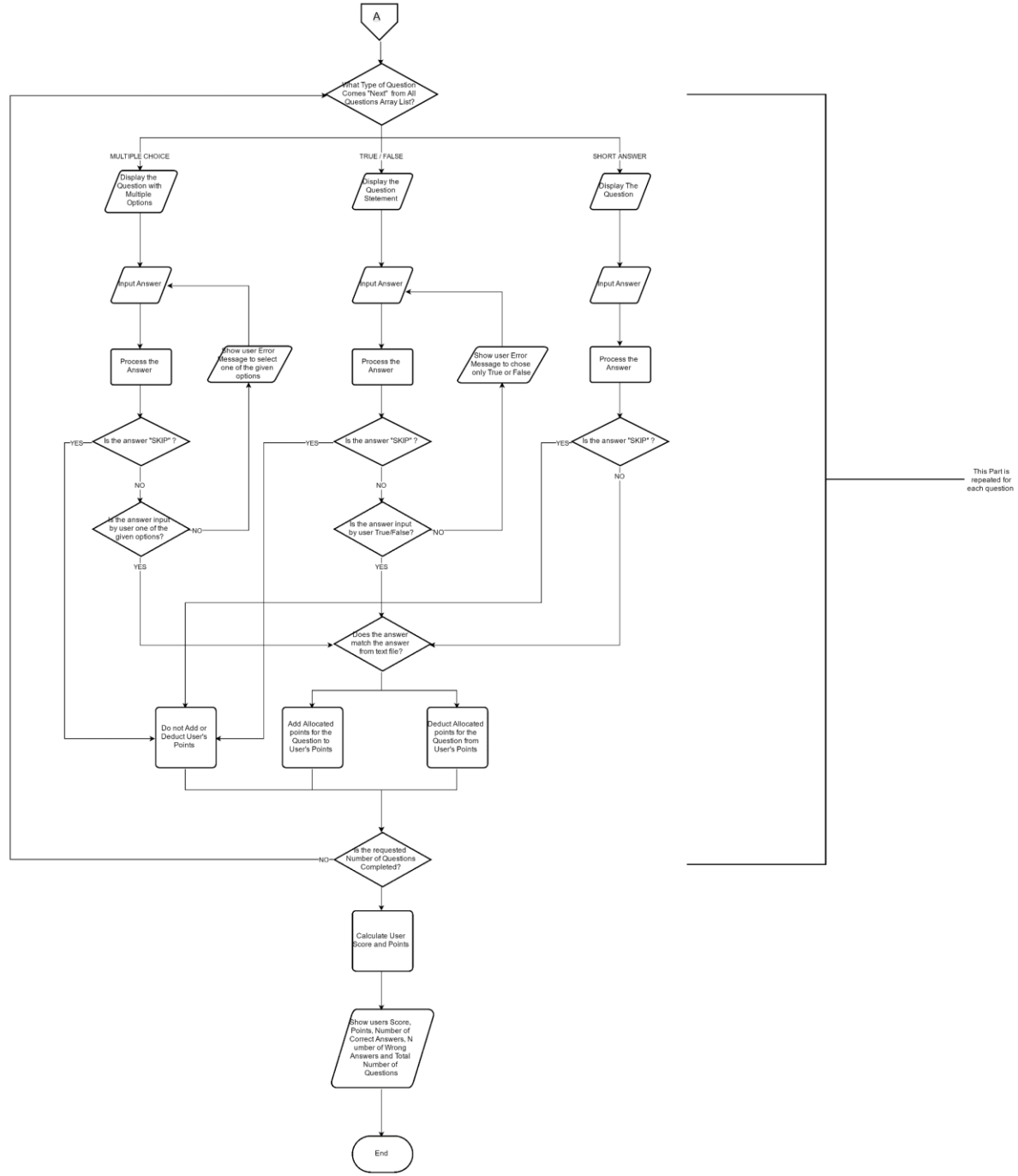
A massege displayed in JLabel telling the user that the question is added



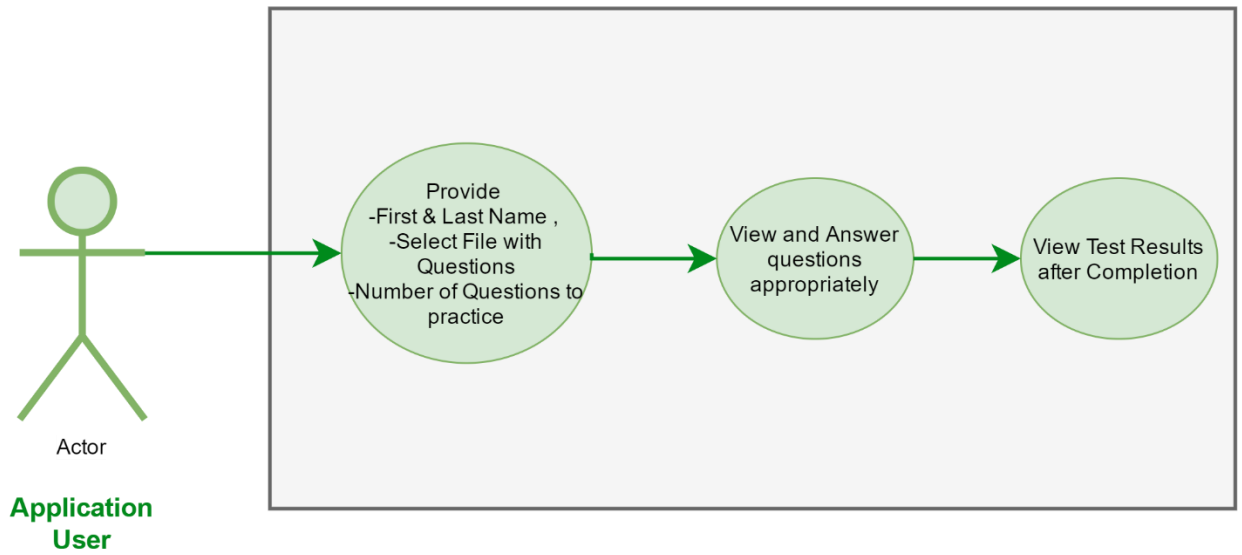
An error massege displayed in the JLabel

QUIZ BOWL


Start
Launch the Application
Input First Name
Input Last Name
Chose File
Process The File
Show File Format Error
Does the File Follow Question Format
NO
YES
Count the Number of Questions without Dubplicates
Show Number of Questions Requested lees than 1 Error
Input Number of Questions to Practice
Show Number of Questions Avaiable in the File
YES
NO
Is number of questions requested is less than 1
Process the Input Number of Questions
Is number of questions requested is available in the text
NO
YES
Process The Types of Questions in File
Show Welcome Message
A


A
What Type of Question Comes "Next" from All Questions Array List?
MULTIPLE CHOICE
TRUE / FALSE
SHORT ANSWER
Display the Question with Multiple Options
Display the Question Stetement
Display The Question
Input Answer
Input Answer
Input Answer
Process the Answer
Show user Error Message to select one of the given options
Process the Answer
Show user Error Message to chose only True or False
Process the Answer
Is the answer "SKIP" ?
YES
NO
Is the answer "SKIP" ?
YES
NO
Is the answer "SKIP" ?
YES
NO
Is the answer input by user one of the given options?
NO
YES
Is the answer input by user True/False?
NO
YES
Does the answer match the answer from text file?
Do not Add or Deduct User's Points
Add Allocated points for the Question to User's Points
Deduct Allocated points for the Question from User's Points
Is the requested Number of Questions Completed?
NO
Calculate User Score and Points
Show users Score, Points, Number of Correct Answers, N umber of Wrong Answers and Total Number of Questions
End
This Part is repeated for each question

QuizBowl Application
Use Case Diagram



Application User

QUIZ BOWL

CODE

```java
public static void main(String[] arg) {
    try {
        // JTatoo library is used for Look and Feel of the program
        UIManager.setLookAndFeel("com.jtattoo.plaf.aluminium.AluminiumLookAndFeel");
    } catch (ClassNotFoundException | InstantiationException | IllegalAccessException ex) {
        System.out.println("Error: " + ex);
    }
    // Calling QuizBowl
    new QuizBowl();
}
```

```java
public static String[] getFileContent(File f) throws FileNotFoundException {
     // Method to get file contents

    String con = "";
    Scanner sc = new Scanner(f);
    if (sc.hasNextLine()) {
        sc.nextLine();
    }
    while (sc.hasNextLine()) { // While the text file has next line it is imported into app
        con = con + sc.nextLine() + "\n";
    }
    return con.split("\n--------\n");   // Splitting the question segments by 8 minuses

}
```

```java
fileButton.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {

        if (fileChooser.showOpenDialog(null) == JFileChooser.APPROVE_OPTION) {
            file = fileChooser.getSelectedFile();
            fileLoc = file.getAbsolutePath();
            submit.setVisible(true);
        }

    }

});
```

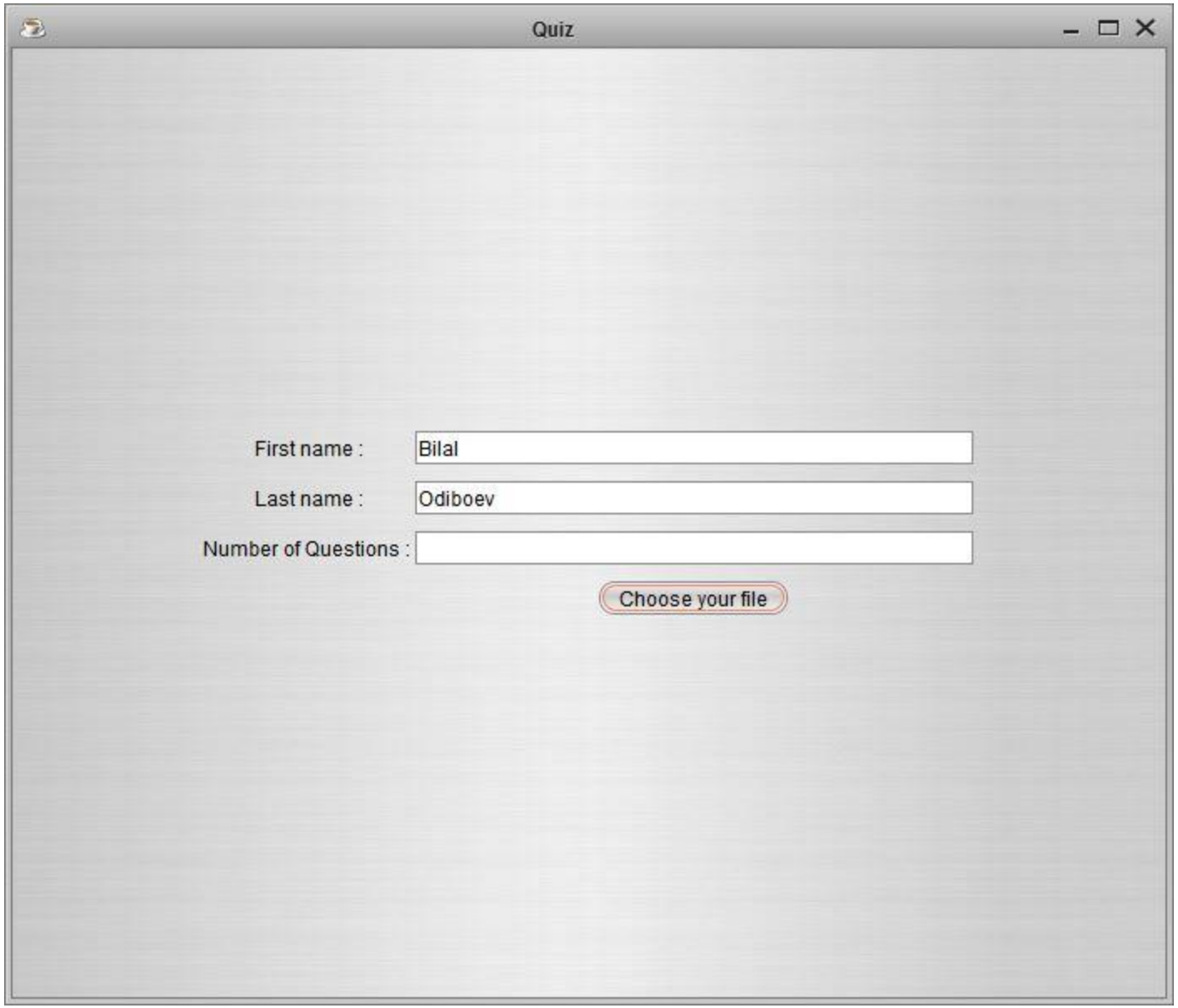
```
// Splitting the contents of text file into elements by splitting them at line breaks
                        for (int I = 0; I < content.length; I++) {
                            String qContent = content[I];
                            String[] qInfo = qContent.split("\n");
                            String qText = qInfo[2];
                            String correct = qInfo[qInfo.length - 1];
                            int points = Integer.parseInt(qInfo[0]);

// Extracting points from first line of text file and storing as integer points through parsing method
                            if (checkDuplicate(allQuestions, qText)) {
// CheckDuplicate method is called to check duplicate questions
                                if (qInfo[1].equals("MC")) {
                                    int numbOfChoices = Integer.parseInt(qInfo[3]);
// Extracting number of choices from 3rd line of question text
                                    String choicesStr = "";
                                    for (int i = 4; i < numbOfChoices + 4; i++) {
                                        choicesStr = choicesStr + qInfo[i] + "\n";
                                    }
                                    allQuestions.add(new QuestionMC(choicesStr.split("\n"), qText, points, correct));
// Calling QuestionMC Class
                                } else if (qInfo[1].equals("TF")) {
                                    allQuestions.add(new QuestionTF(qText, points, correct));
// Calling QuestionTF Class
                                } else if (qInfo[1].equals("SA")) {
                                    allQuestions.add(new QuestionSA(qText, points, correct));
// Calling QuestionSA Class
                                }
                            }
                        }
```

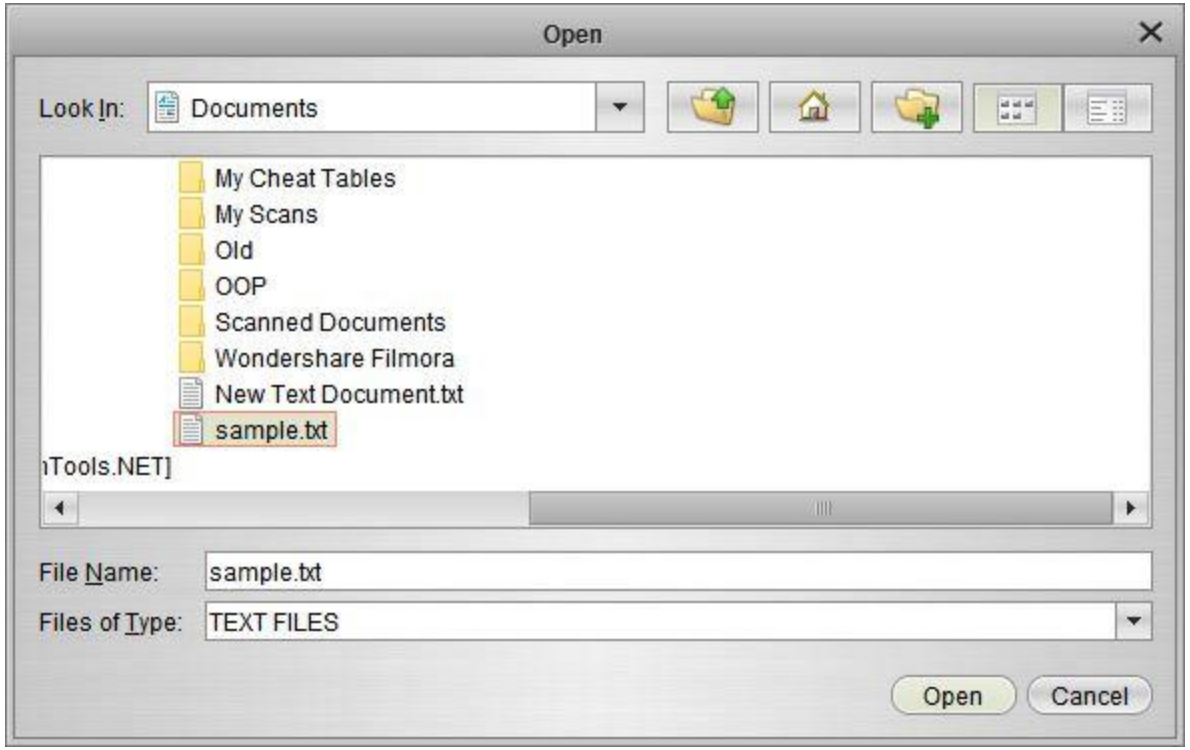
```java
// answer button ("Next" Button) action listener
        answer.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                Object a = allQuestions.get(qSerialNumber);       // Assigning a number to each question from the text
                if (a.getClass().getName().equals("QuestionMC")) {
                    if (((QuestionMC) a).Check(answerField.getText())) {      // Calling QuestionMC and Checking if user input answer is correct
                        gamer.setPoints(gamer.getPoints() + ((QuestionMC) a).answer(answerField.getText()));
                        qSerialNumber++;       // Iterating through questions in text file
                        result.setText("");
                        cal(((QuestionMC) a).answer(answerField.getText()));
                    } else {
                        result.setText("<html><h3 style='color:red'>You have to choose one of the choices</h3></html>");   // ERROR MESSAGE IN HTML
                    }
                } else if (a.getClass().getName().equals("QuestionTF")) {
                    if (((QuestionTF) a).Check(answerField.getText())) {
                        gamer.setPoints(gamer.getPoints() + ((QuestionTF) a).answer(answerField.getText()));   // Calling QuestionTF Class
                        qSerialNumber++;
                        cal(((QuestionTF) a).answer(answerField.getText()));    // Iterating through questions in text file
                        result.setText("");
                    } else {
                        result.setText("<html><h3 style='color:red'>You answer by (True or False)</h3></html>");
                    }
                } else if (a.getClass().getName().equals("QuestionSA")) {
                    gamer.setPoints(gamer.getPoints() + ((QuestionSA) a).answer(answerField.getText()));
                    qSerialNumber++;
                    result.setText("");
                    cal(((QuestionSA) a).answer(answerField.getText()));
                }
```

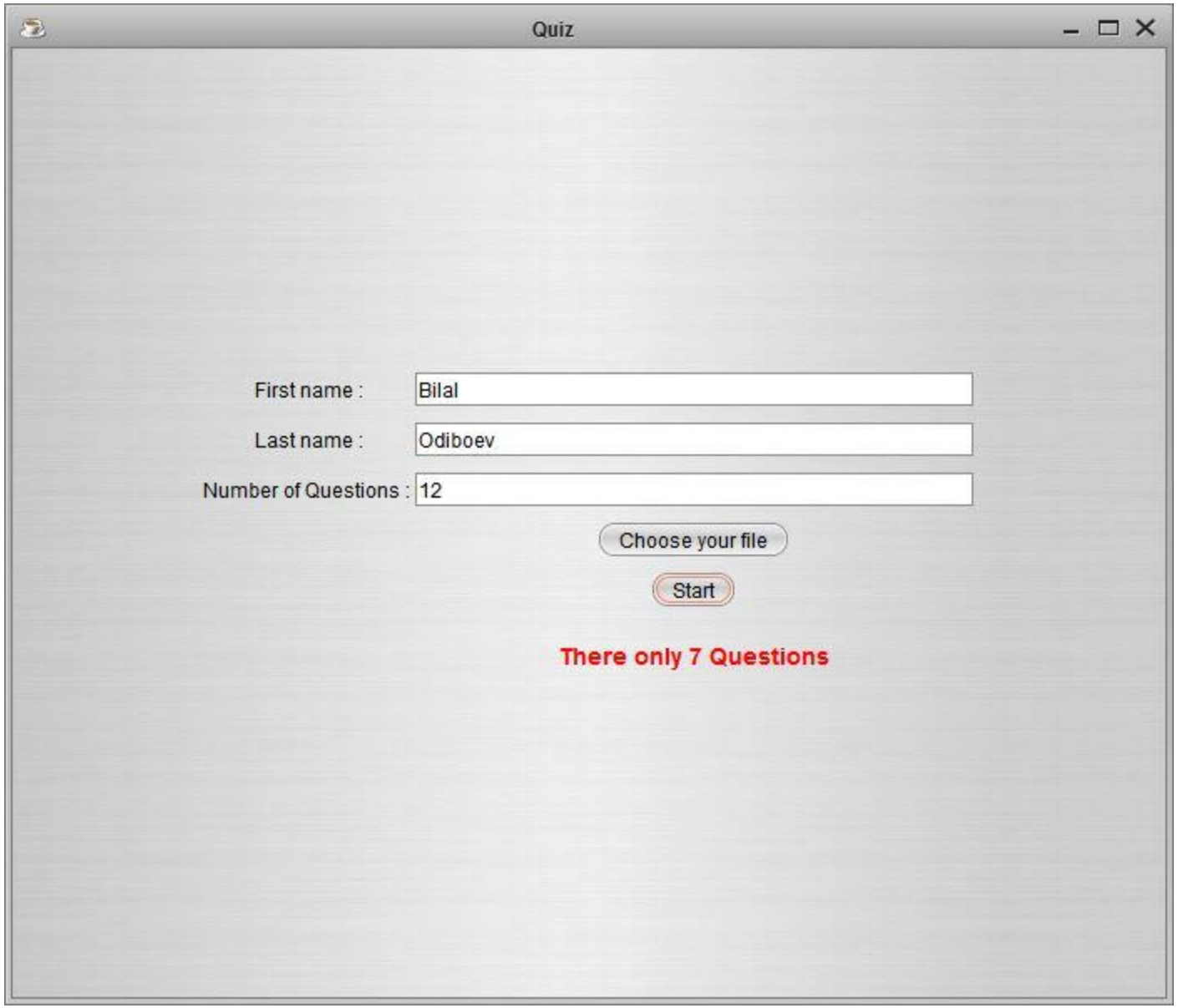
```java
// Once the Quiz has Ended

if (qSerialNumber == NumReqQuestions || qSerialNumber >= QuesFileNumb) {
                                System.out.println(gamer.getPoints());
    if (gamer.getPoints() < 0) {
        gamer.setPoints(0);
    }
    gamerLabel.setText(helloStranger + "<br>Points : " + gamer.getPoints());
    qSerialNumber++;
    Question.setVisible(false);
    answer.setVisible(false);
    answerField.setVisible(false);
    answerLabel.setVisible(false);
    gamerLabel.setVisible(false);
    FinalLabel.setText("<html><h3>" + helloStranger
          + "<br>Your points : " + gamer.getPoints()
          + "<br>Number of correct answers : " + answerdCorrectly
          + "<br>Number of wrong answers : " + answerdWrongly
          + "<br> Skipped Questions : " + skipped
          + "<br><hr><br>Total : " + NumReqQuestions
          + "<br>" + "OverAll : " + (100 * answerdCorrectly / NumReqQuestions) + "%");

          // Calculating the percentage of corrct answers
```

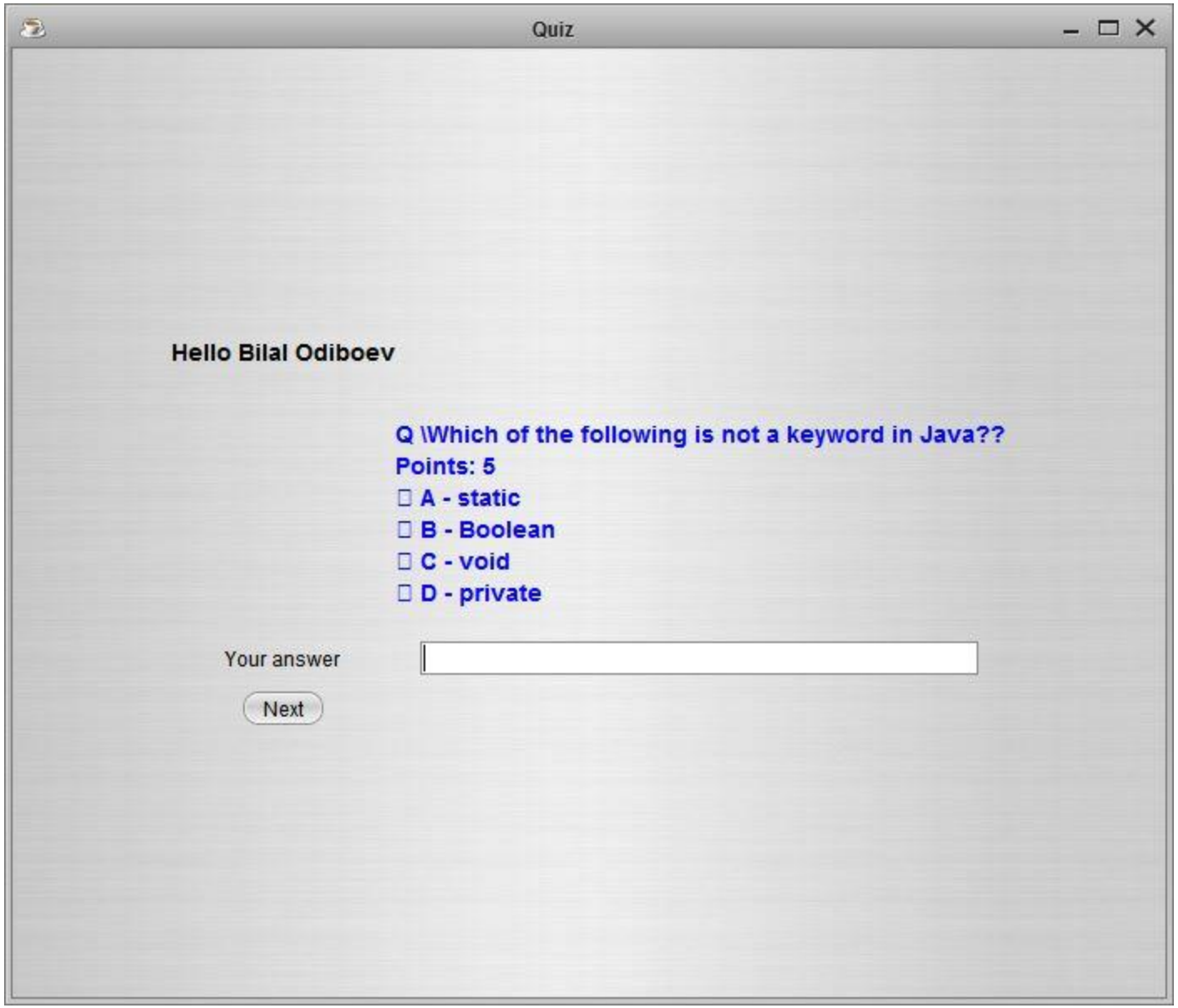
QUIZ BOWL

UI

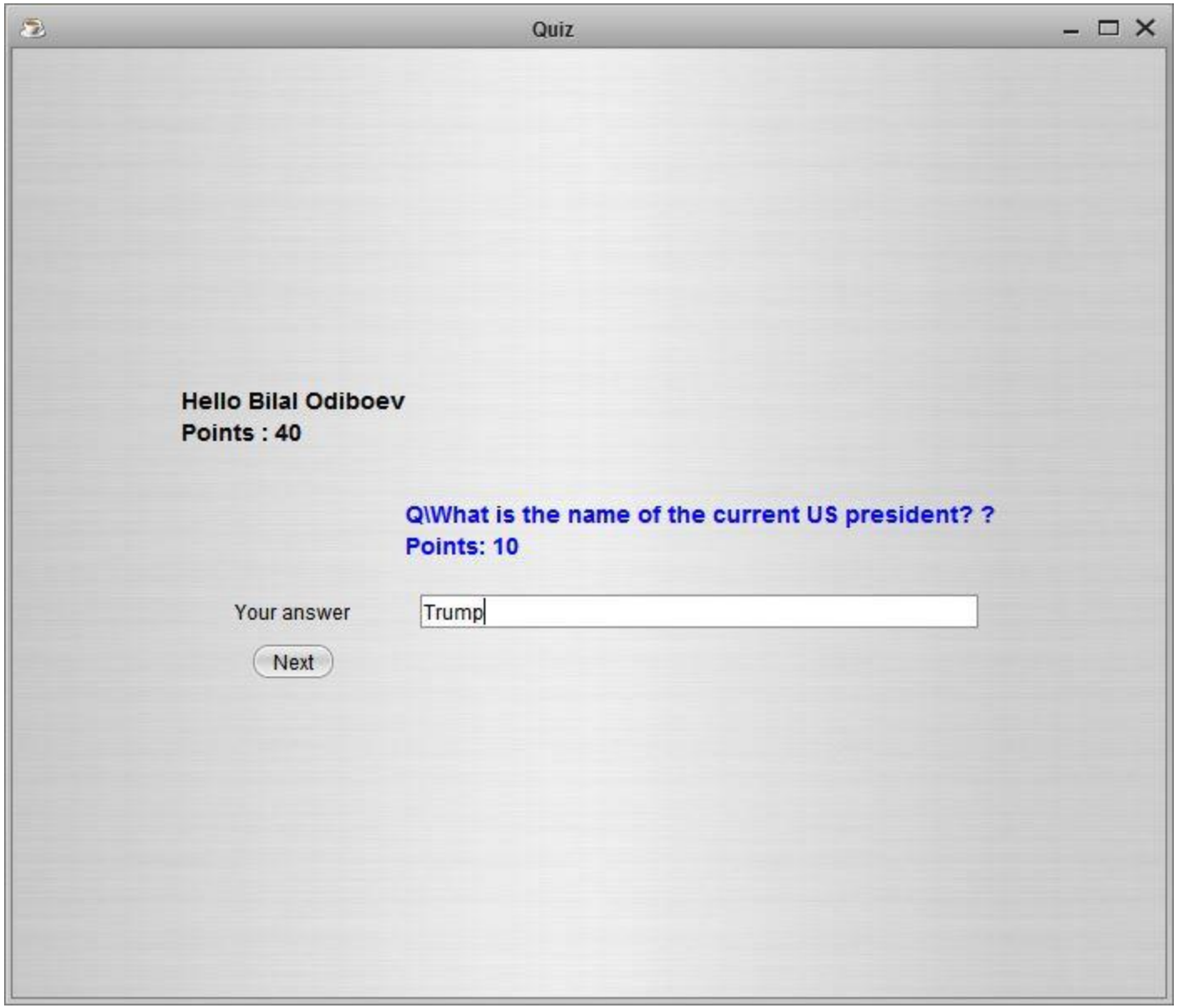

Quiz
First name :
Bilal
Last name :
Odiboev
Number of Questions :
Choose your file

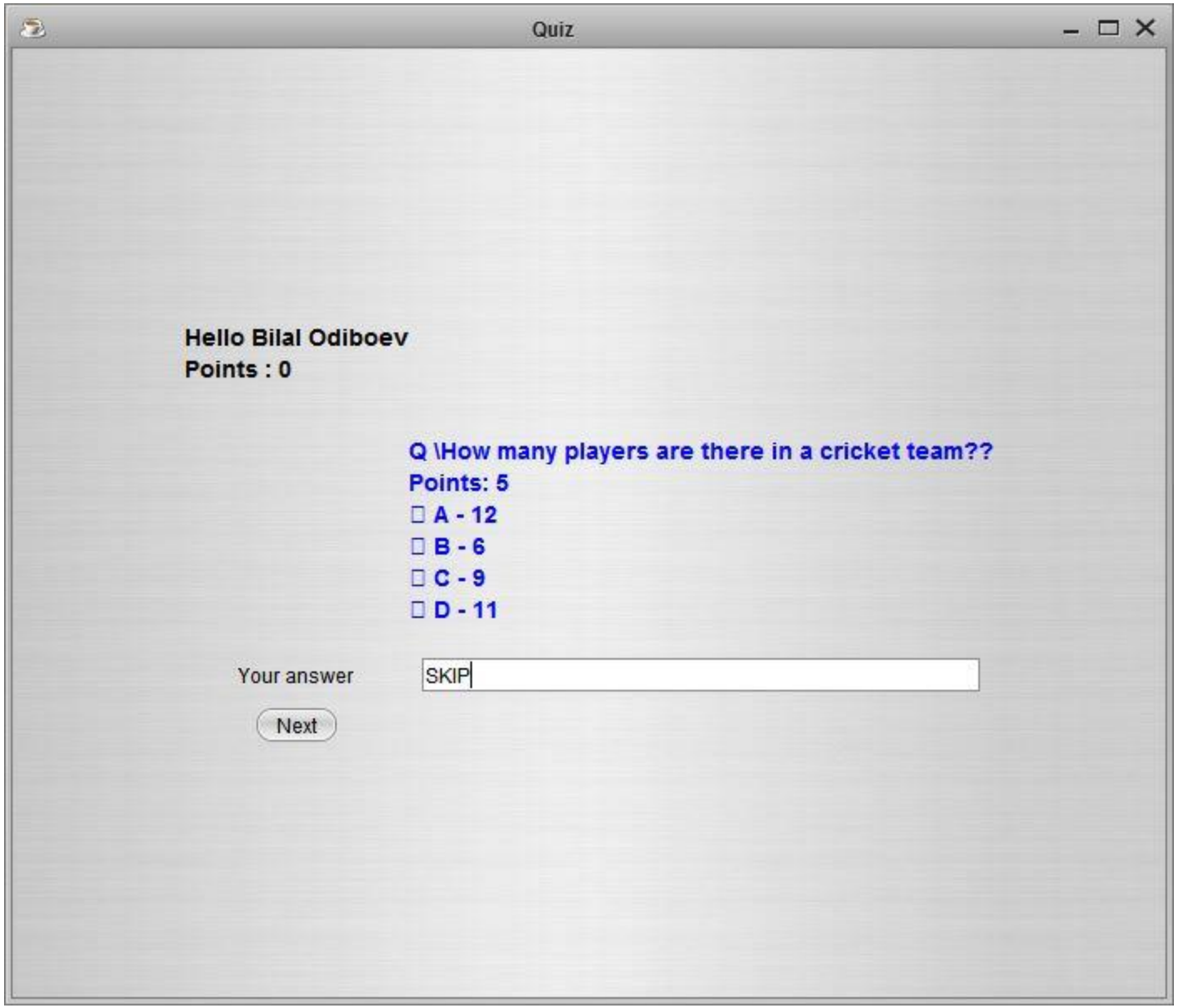

Open
Look In:
Documents
My Cheat Tables
My Scans
Old
OOP
Scanned Documents
Wondershare Filmora
New Text Document.txt
sample.txt
ıTools.NET]
File Name:
sample.txt
Files of Type:
TEXT FILES
Open
Cancel

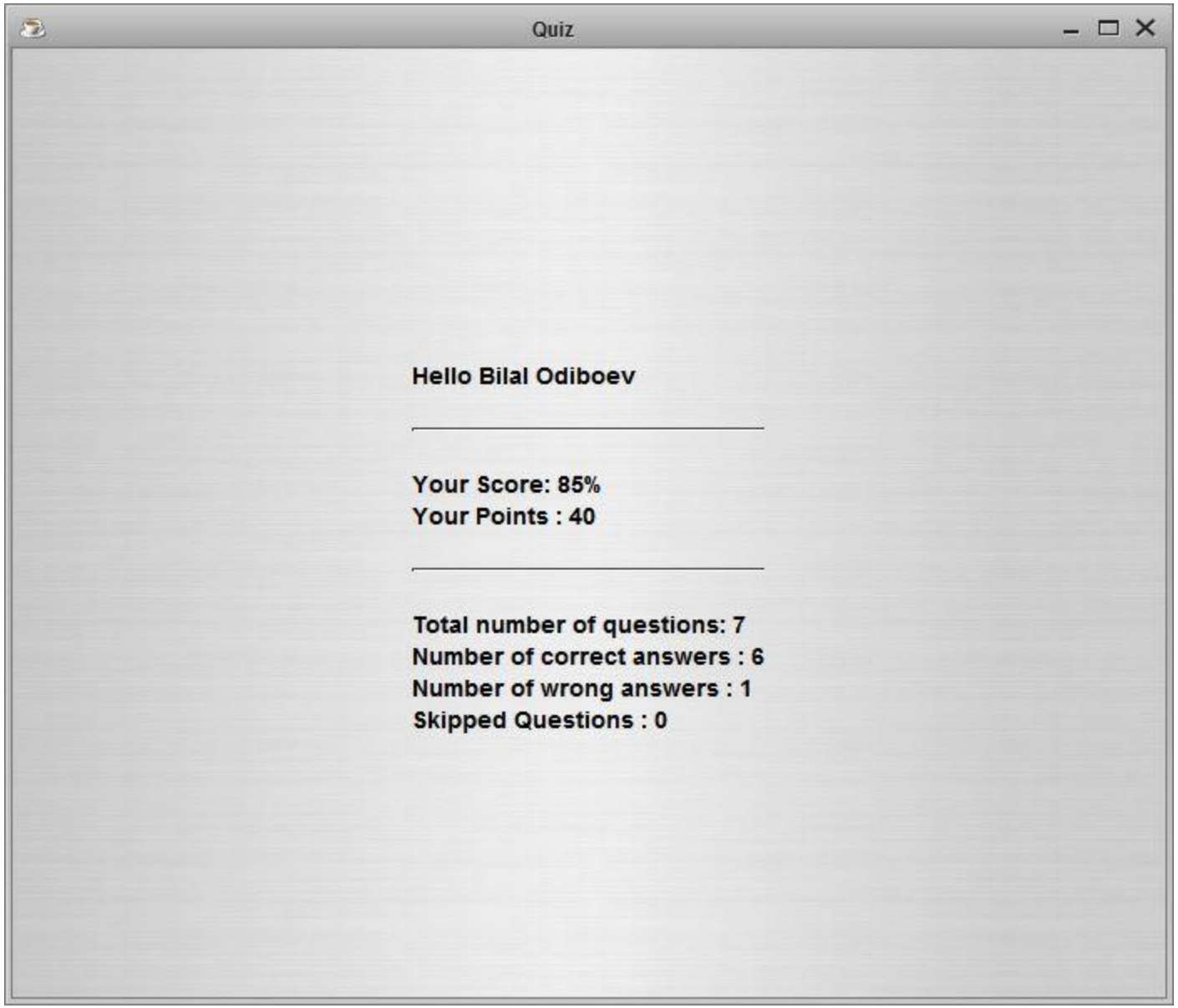

Quiz
First name :
Bilal
Last name :
Odiboev
Number of Questions :
12
Choose your file
Start
There only 7 Questions


Quiz
Hello Bilal Odiboev
Q \Which of the following is not a keyword in Java??
Points: 5
A - static
B - Boolean
C - void
D - private
Your answer
Next


Quiz
Hello Bilal Odiboev
Points : 10
Q\Was Java created in 1995??? (Answer by True or False)
Points: 10
Your answer
True
Next


Quiz
Hello Bilal Odiboev
Points : 40
Q\What is the name of the current US president? ?
Points: 10
Your answer
Trump
Next


Quiz
Hello Bilal Odiboev
Points : 0
Q \How many players are there in a cricket team??
Points: 5
A - 12
B - 6
C - 9
D - 11
Your answer
SKIP
Next


Quiz
Hello Bilal Odiboev
Your Score: 85%
Your Points : 40
Total number of questions: 7
Number of correct answers : 6
Number of wrong answers : 1
Skipped Questions : 0

Conclusion

- In conclusion we have developed two application that is the Quiz bowl application and the Question Creation Application.
- We utilizes multiple classes and inheritance in our application.

Thank You For Listening To Our Presentation