**Department of Computer Science & Engineering, SDMCET, Dharwad-2**


# AOOP Assignment Submission Report

**[Submitted as part of CTA Assignment No-2]**

| | | | |
|---|---|---|---|
| Course: | Advanced Object-Oriented Programming | Course Code: | 18UCSE508 |
| Semester: | V | Division: | B |

Submitted by:



| | | | |
|---|---|---|---|
| USN: | 2SD20CS121 | Name: | VENUGOPAL K MALLI |

Q1. Write a Java program to build the GUI application using JavaFX for the following requirements:

a) Read user name and password using appropriate JavaFX controls.

b) Validate the input. If user name and password are matched with the assumed values, then display the welcome scene with proper text.

c) If user name and password don't match, then raise appropriate exception

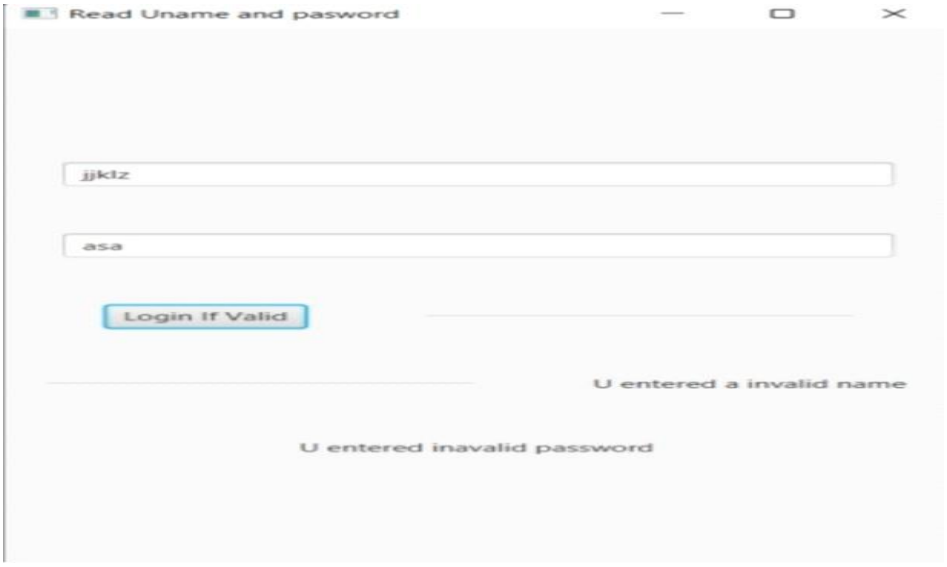
**PROGRAM:**

```
package application;

import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.geometry.Insets;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.PasswordField;
import javafx.scene.control.TextField;
import javafx.scene.layout.GridPane;
import javafx.stage.Stage;


public class Q1 extends Application {
	String usrNameString= "Sumanth";
	String passwordString="1234";

	@Override
	public void start(Stage primaryStage) {
		try {
			GridPane root = new GridPane();
```

```
                root.setHgap(20);
                root.setVgap(20);
                root.setPadding(new Insets(0,10,0,10));
                Scene scene = new Scene(root,300,200);

        scene.getStylesheets().add(getClass().getResource("application.css").toExternalForm());
                primaryStage.setScene(scene);
                root.setGridLinesVisible(false);

                Label usrNameLabel = new Label("User Name");
                root.add(usrNameLabel, 0, 1);
                Label passWdLabel = new Label("Password");
                root.add(passWdLabel, 0, 2);
                Label responseLabel = new Label();
                root.add(responseLabel, 2, 4);

                TextField usrnNameTextField = new TextField();
                root.add(usrnNameTextField, 2, 1);
                PasswordField passwordField = new PasswordField();
                root.add(passwordField, 2, 2);

                Button suButton = new Button("Submit");

                root.add(suButton, 2, 3);

                suButton.setOnAction(new EventHandler<ActionEvent>() {
                        @Override
                        public void handle(ActionEvent arg0) {
                                String str1 = usrnNameTextField.getText();
                                String str2 = passwordField.getText();

                                try {

        if(str1.equalsIgnoreCase(usrNameString)&&str2.equals(passwordString)) {
                                                        responseLabel.setText("Login
Succesful");
```

```
                                        }else {
                                                throw new
InvalidUserNameOrPasswordException();
                                        }
                                    } catch (InvalidUserNameOrPasswordException e) {
                                        System.out.println(e);
                                    }

                                }
                            });

//root.getChildren().addAll(usrNameLabel,usrnNameTextField,passWdLabel,passwordField,suB
utton,responseLabel);

                            primaryStage.show();

                        } catch(Exception e) {
                            System.err.println(e);
                        }
                }

                public static void main(String[] args) {
                        launch(args);
                }
}

class InvalidUserNameOrPasswordException extends Exception{

                private static final long serialVersionUID = 1L;

                public InvalidUserNameOrPasswordException() {
                        super("Invalid UserName or Password!!");
                }
}
```

# EXECUTION:


Read Uname and pasword
jjkdz
asa
Login If Valid
U entered a invalid name
U entered inavalid password

Q2. Write a Java program to build the GUI application using JavaFX for the following requirements:

a) Create a Menu control to display the menu items: File, Edit & Help.

b) Create sub menus in the order: File → New, Open & Save. Edit → Cut, Copy & Paste.

Help → Help Centre, About Us

The program must use Mnemonics and Accelerators (wherever appropriate) to Menu Items.

## PROGRAM:

```
package application;


import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.geometry.Insets;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.scene.control.Menu;
import javafx.scene.control.MenuBar;
import javafx.scene.control.MenuItem;
import javafx.scene.input.KeyCombination;
import javafx.scene.layout.BorderPane;
import javafx.stage.Stage;


public class Q2 extends Application {
	String usrNameString= "Sumanth";
	String passwordString="1234";

	@Override
```

```
        public void start(Stage primaryStage) {
                try {

                        primaryStage.setTitle("Menu Design");

                        BorderPane root = new BorderPane();

                        root.setPadding(new Insets(0,0,0,0));
                        Scene scene = new Scene(root,300,200);

        scene.getStylesheets().add(getClass().getResource("application.css").toExt
ernalForm());
                        primaryStage.setScene(scene);



                        MenuBar menuBar = new MenuBar();

                        Menu fileMenu = new Menu("File");


                        MenuItem fnewMenuItem = new MenuItem("New");
                        MenuItem fopenMenuItem = new MenuItem("Open");
                        MenuItem fsaveMenuItem = new MenuItem("Save");

                        fileMenu.setMnemonicParsing(true);


        fnewMenuItem.setAccelerator(KeyCombination.keyCombination("shortcut
+n"));
```

```
        fopenMenuItem.setAccelerator(KeyCombination.keyCombination("shortcu
t+o"));

        fsaveMenuItem.setAccelerator(KeyCombination.keyCombination("shortcut
+s"));


        fileMenu.getItems().addAll(fnewMenuItem,fopenMenuItem,fsaveMenuIte
m);

                menuBar.getMenus().add(fileMenu);

                Menu editMenu = new Menu("Edit");

                MenuItem cutMenuItem = new MenuItem("Cut");
                MenuItem copyMenuItem = new MenuItem("Copy");
                MenuItem pasteMenuItem = new MenuItem("Paste");

                editMenu.setMnemonicParsing(true);


        cutMenuItem.setAccelerator(KeyCombination.keyCombination("shortcut+
x"));

        copyMenuItem.setAccelerator(KeyCombination.keyCombination("shortcut
+c"));

        pasteMenuItem.setAccelerator(KeyCombination.keyCombination("shortcut
+v"));
```

```
        editMenu.getItems().addAll(cutMenuItem,copyMenuItem,pasteMenuItem
);
                    menuBar.getMenus().add(editMenu);

                    Menu helpMenu = new Menu("Help");

                    MenuItem helpcentreMenuItem = new MenuItem("Help
Centre");
                    MenuItem aboutMenuItem = new MenuItem("About Us");

                    helpMenu.setMnemonicParsing(true);


        helpcentreMenuItem.setAccelerator(KeyCombination.keyCombination("F1
"));



        helpMenu.getItems().addAll(helpcentreMenuItem,aboutMenuItem);
                    menuBar.getMenus().add(helpMenu);

                    Label response = new Label();

                    root.setTop(menuBar);
                    root.setCenter(response);

                    EventHandler<ActionEvent> menuEventHandler = new
EventHandler<ActionEvent>() {
                            @Override
                            public void handle(ActionEvent ae) {
```

```
                                String outputString =
((MenuItem)ae.getTarget()).getText();
                                if(outputString.equals("Exit")) {
                                        System.exit(0);
                                }

                                response.setText(outputString + " selected");

                        }
                };

                fnewMenuItem.setOnAction(menuEventHandler);
                fopenMenuItem.setOnAction(menuEventHandler);
                fsaveMenuItem.setOnAction(menuEventHandler);
                cutMenuItem.setOnAction(menuEventHandler);
                copyMenuItem.setOnAction(menuEventHandler);
                pasteMenuItem.setOnAction(menuEventHandler);
                helpcentreMenuItem.setOnAction(menuEventHandler);
                aboutMenuItem.setOnAction(menuEventHandler);



                primaryStage.show();
        } catch(Exception e) {
                System.err.println(e);
        }
    }
    public static void main(String[] args) {
        launch(args);
    }
}
```

Q3. Write a Java program to build the GUI application using JavaFX for the following requirements:

a) Create Context menu involving the menu items in the order: New & View.

b) Create sub menus for the above main context menu: New → File, Folder & Image.

View → Large, Medium & Small.

The context menu must be displayed on right-click of the mouse button.

## PROGRAM:

```
package application;



import javafx.application.Application;

import javafx.event.ActionEvent;

import javafx.event.EventHandler;

import javafx.geometry.Insets;

import javafx.geometry.Pos;

import javafx.scene.Scene;

import javafx.scene.control.ContextMenu;

import javafx.scene.control.Label;
```

```
import javafx.scene.control.Menu;

import javafx.scene.control.MenuItem;

import javafx.scene.control.TextField;

import javafx.scene.layout.FlowPane;

import javafx.stage.Stage;



public class Q3 extends Application {

        String usrNameString= "Sumanth";

        String passwordString="8765";


        @Override

        public void start(Stage primaryStage) {

                try {


                        primaryStage.setTitle("Menu Design");


                        FlowPane root = new FlowPane(10,10);

                        root.setAlignment(Pos.CENTER);
```

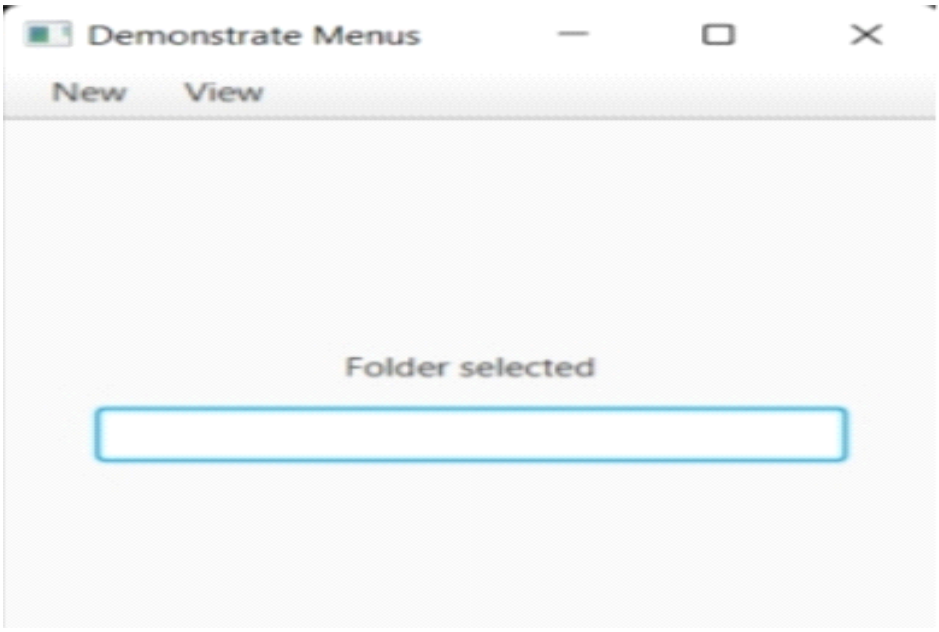
```
                root.setPadding(new Insets(0,0,0,0));

                Scene scene = new Scene(root,300,200);


        scene.getStylesheets().add(getClass().getResource("application.css").toExt
ernalForm());

                primaryStage.setScene(scene);


                Label responseLabel = new Label();




                Menu newMenu = new Menu("New");


                MenuItem fileMenuItem = new MenuItem("File");

                MenuItem folderMenuItem = new MenuItem("Folder");

                MenuItem imageMenuItem = new MenuItem("Image");


        newMenu.getItems().addAll(fileMenuItem,folderMenuItem,imageMenuIte
m);
```

```
        Menu viewMenu = new Menu("View");


        MenuItem largeMenuItem = new MenuItem("Large");
        MenuItem mediuMenuItem = new MenuItem("Medium");
        MenuItem smallMenuItem = new MenuItem("Small");


    viewMenu.getItems().addAll(largeMenuItem,mediuMenuItem,smallMenuIt
em);

        ContextMenu contextMenu = new ContextMenu();
         contextMenu.getItems().addAll(newMenu,viewMenu);


        TextField txTextField = new TextField();


        txTextField.setPrefColumnCount(20);
        txTextField.setContextMenu(contextMenu);
```

```
                EventHandler<ActionEvent> meEventHandler = new
EventHandler<ActionEvent>() {
                    @Override
                    public void handle(ActionEvent ae){
                        String name = ((MenuItem)
ae.getTarget()).getText();

                        responseLabel.setText(name +" selected");
                    }
                };


                fileMenuItem.setOnAction(meEventHandler);
                folderMenuItem.setOnAction(meEventHandler);
                imageMenuItem.setOnAction(meEventHandler);
                largeMenuItem.setOnAction(meEventHandler);
                mediuMenuItem.setOnAction(meEventHandler);
                smallMenuItem.setOnAction(meEventHandler);
```

```
                root.getChildren().addAll(txTextField,responseLabel);


                primaryStage.show();
            } catch(Exception e) {
                System.err.println(e);
            }
        }

        public static void main(String[] args) {
            launch(args);
        }
}
```

**OUTPUT:**



Q4. Write a JavaFX program that produces the following output when executed and displays Dialog Box

(as shown in Figure.2) on click of Register button (as shown in Figure.1):


Figure.1

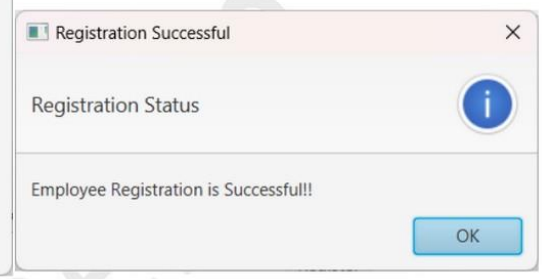



Figure. 2

**PROGRAM:**

```
package application;



import javafx.application.Application;

import javafx.geometry.Insets;

import javafx.geometry.Pos;

import javafx.scene.Scene;

import javafx.scene.control.Alert;

import javafx.scene.control.Alert.AlertType;
```

```
import javafx.scene.control.Button;
import javafx.scene.control.CheckBox;
import javafx.scene.control.ComboBox;
import javafx.scene.control.DatePicker;
import  javafx.scene.control.Label;
import javafx.scene.control.RadioButton;
import javafx.scene.control.TextField;
import javafx.scene.control.ToggleGroup;
import javafx.scene.layout.BorderPane;
import javafx.scene.layout.HBox;
import javafx.scene.layout.VBox;
import javafx.scene.text.Font;
import javafx.stage.Stage;



public class Q4 extends Application {
        String usrNameString= "Sumanth";
        String passwordString="8765";

        @Override
        public void start(Stage primaryStage) {
                try {
```

```
                VBox root = new VBox();
                root.setAlignment(Pos.CENTER);
                root.setSpacing(10);
                root.setPadding(new Insets(0,10,0,50));
                Scene scene = new Scene(root,500,300);

        scene.getStylesheets().add(getClass().getResource("application.css").toExt
ernalForm());
                primaryStage.setScene(scene);


                Label headingLabel = new Label("Employee Registration
Form");
                headingLabel.setFont(Font.font(22));
                BorderPane.setAlignment(headingLabel, Pos.CENTER);

                Label nameLabel = new Label("Enter Your Name:");


                TextField nameField = new TextField();
                nameField.setPromptText("Enter Your Name");
                nameField.setFocusTraversable(false);
```

```
                HBox nameBox = new HBox();
                nameBox.setSpacing(10);
                nameBox.getChildren().addAll(nameLabel,nameField);

                Label genderLabel = new Label("Enter Your Gender");
                RadioButton maleButton = new RadioButton("Male");
                RadioButton femaleButton = new RadioButton("Female");

                ToggleGroup genderGroup = new ToggleGroup();
                maleButton.setToggleGroup(genderGroup);
                femaleButton.setToggleGroup(genderGroup);

                HBox genderBox = new HBox();
//              genderBox.setPadding(new Insets(10));
                genderBox.setSpacing(10);
                genderBox.getChildren().addAll(genderLabel,
maleButton,femaleButton);



                Label dobLabel = new Label("Enter Date of Birth");
                DatePicker dobDatePicker = new DatePicker();
```

```
                HBox dateBox = new HBox();
                dateBox.setSpacing(10);
                dateBox.getChildren().addAll(dobLabel,dobDatePicker);

                Label stateLabel = new Label("Select Your State");
                ComboBox<String> stateComboBox = new ComboBox<>();
                stateComboBox.setPrefWidth(130);
                stateComboBox.getItems().addAll("Karnataka","Assam",
"Bihar");

                HBox stateBox =new HBox();
                stateBox.setSpacing(10);
                stateBox.getChildren().addAll(stateLabel,stateComboBox);

                Label qualifiLabel = new Label("Select Your Qualification");
                CheckBox ugCheckBox = new CheckBox("UG");
                CheckBox pgCheckBox = new CheckBox("PG");
                CheckBox phdCheckBox = new CheckBox("PhD");

                HBox qualifiBox = new HBox();
```

```
                qualifiBox.setSpacing(10);

                qualifiBox.getChildren().addAll(qualifiLabel,
ugCheckBox,pgCheckBox,phdCheckBox);



                Button registerButton = new Button("Register");



                registerButton.setOnAction((ae)->{



                        Alert registerDialog = new
Alert(AlertType.INFORMATION);

                        registerDialog.setTitle("Registration Successful");



                        registerDialog.setHeaderText("Registration Status");

                        registerDialog.setContentText("Employee Registration
Successful!!");

                        registerDialog.showAndWait();



                } );;









        root.getChildren().addAll(headingLabel,nameBox,genderBox,dateBox,state
Box,qualifiBox,registerButton);
```

```
                primaryStage.show();


            } catch(Exception e) {
                System.err.println(e);
            }
        }

        public static void main(String[] args) {
            launch(args);
        }
}
```

**OUTPUT:**

## 3. Screen Shots of Execution: