*CSE 102*

**JavaFX**

* So far any interactive program we have ever written interacts the user through what is called a CLI (command-line interface). However, almost all modern programs have a GUI (graphical-user interface) component which offers more sophisticated ways of interaction (clicking buttons, scrolling, drag-n-drop etc.).
* JavaFX is a modern Java library for writing GUI programs in an easy way. It has lots and lots of built-in classes, methods and constants to reduce the effort needed to develop an app. Having atomic problems already solved, we can focus our attention on the program logic and design.
* [Here](https://math.hws.edu/eck/cs124/javanotes9/c6/index.html) is one of the simple tutorials that you can follow to get your hands dirty with your first GUI app.

**Exercises:**

1. Which of the following cannot be the root node of a scene graph in JavaFX?
2. GridPane
3. Canvas
4. HBox
5. VBox
6. How does a BorderPane arrange the components that it contains?
7. Horizontally next to each other
8. Vertically next to each other
9. On top of each other
10. Center, right, left, top and bottom
11. Which of the following image formats is not supported by the Image class?
12. PNG
13. SVG
14. JPEG
15. GIF
16. Which of the following is not a valid way of creating a Color object?
17. Color c = Color.*color*(red, green, blue);
18. Color c = Color.*color*(red, green, blue, opacity);
19. Color c = **new** Color(red, green, blue);
20. Color c = **new** Color(red, green, blue, opacity);
21. Which of the following is the signature of the start method?
22. **public** **void** start(Stage stage)
23. **public** **void** start()
24. **public** **void** start(Scene scene)
25. **public** **void** start(Stage stage, Scene scene)
26. Write an event handler for the following app so that when user enters the correct credentials (stored in the HashMap) and presses the “sign in” button, she is informed with a success message through the “response” label. Furthermore she should be informed through the same label in case she entered wrong credentials.

**public** **void** start(Stage stage) {

Label uname = **new** Label("Username: ");

Label pass = **new** Label("Password: ");

Label response = **new** Label();

Button signIn = **new** Button("Sign in");

TextField unameField = **new** TextField();

PasswordField passField = **new** PasswordField();

GridPane root = **new** GridPane();

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

root.add(uname, 0, 0);

root.add(pass, 0, 1);

root.add(unameField, 1, 0);

root.add(passField, 1, 1);

root.add(signIn, 1, 2);

root.add(response, 1, 3);

HashMap<String, String> userList = **new** HashMap<>();

userList.put("admin", "12345");

signIn.setOnAction( e -> {

// your code

});

Scene scene = **new** Scene(root, 100, 100);

stage.setScene(scene);

stage.setResizable(**false**);

stage.show();

}

1. Write an event handler for the following app so that pressing the button increments the number displayed by the label by one.

**public** **void** start(Stage stage) {

Button button = **new** Button("count");

Label label = **new** Label("0");

VBox root = **new** VBox(label, button);

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

button.setOnAction(e -> {

// your code

});

Scene scene = **new** Scene(root, 100, 100);

stage.setScene(scene);

stage.setResizable(**false**);

stage.show();

}

1. Write a simple app which consists of a single button. A new stage should be created every time the button is pressed.

**ANSWERS:**

1. B
2. D
3. B
4. C
5. A

signIn.setOnAction( e -> {

String unameEntered = unameField.getText();

String passEntered = passField.getText();

**if**(userList.containsKey(unameEntered) &&

userList.get(unameEntered).equals(passEntered))

response.setText("Login successful!");

**else**

response.setText("Access denied!");

});

button.setOnAction(e -> {

**int** cur = Integer.*parseInt*(label.getText());

label.setText(cur + 1 + "");

});

**public** **void** start(Stage stage) {

Button button = **new** Button("new window");

BorderPane root = **new** BorderPane(button);

button.setOnAction(e -> {

**new** Stage().show();

});

Scene scene = **new** Scene(root, 100, 100);

stage.setScene(scene);

stage.show();

}