St Francis Institute of Technology, Mumbai-400 103

Class: SE-ITA/ITB Semester: III; A.Y. 2020-2021
Subject: Java Labs

Title-10: Java Program to implement GUI using JavaFX.

1. **Aim:**
   i. Write a Java program to design a Login Form using JavaFX Controls.

2. **Prerequisite:** Knowledge of AWT,Swings and JavaFX GUI components.

3. **Requirements:** Personal Computer (PC), Windows Operating System, Net beans 8.0.

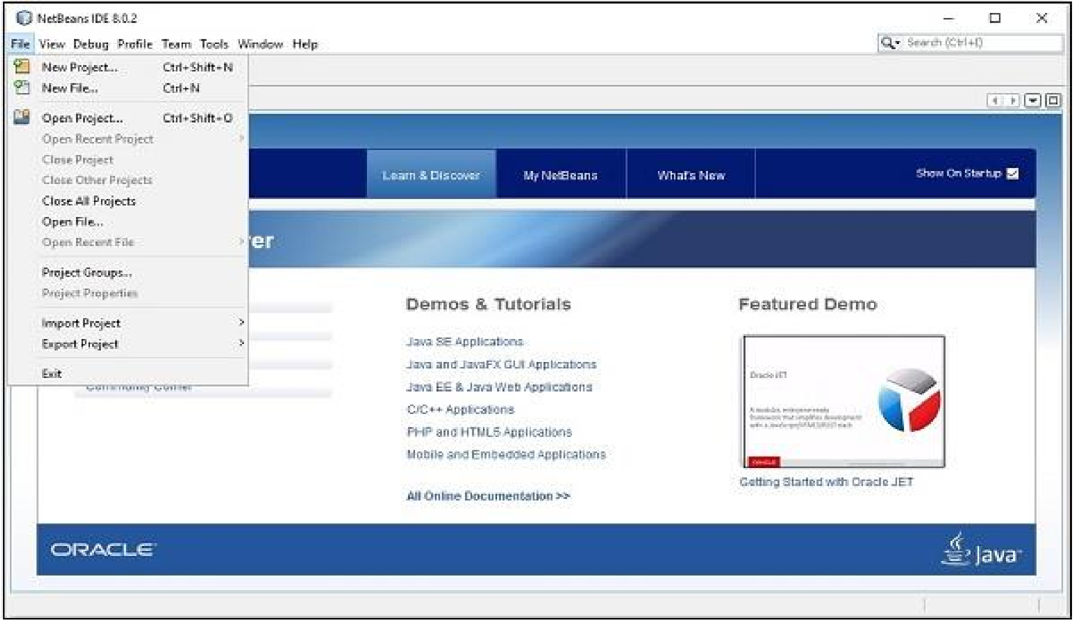
4. **Pre-Experiment Exercise:**
   Theory:

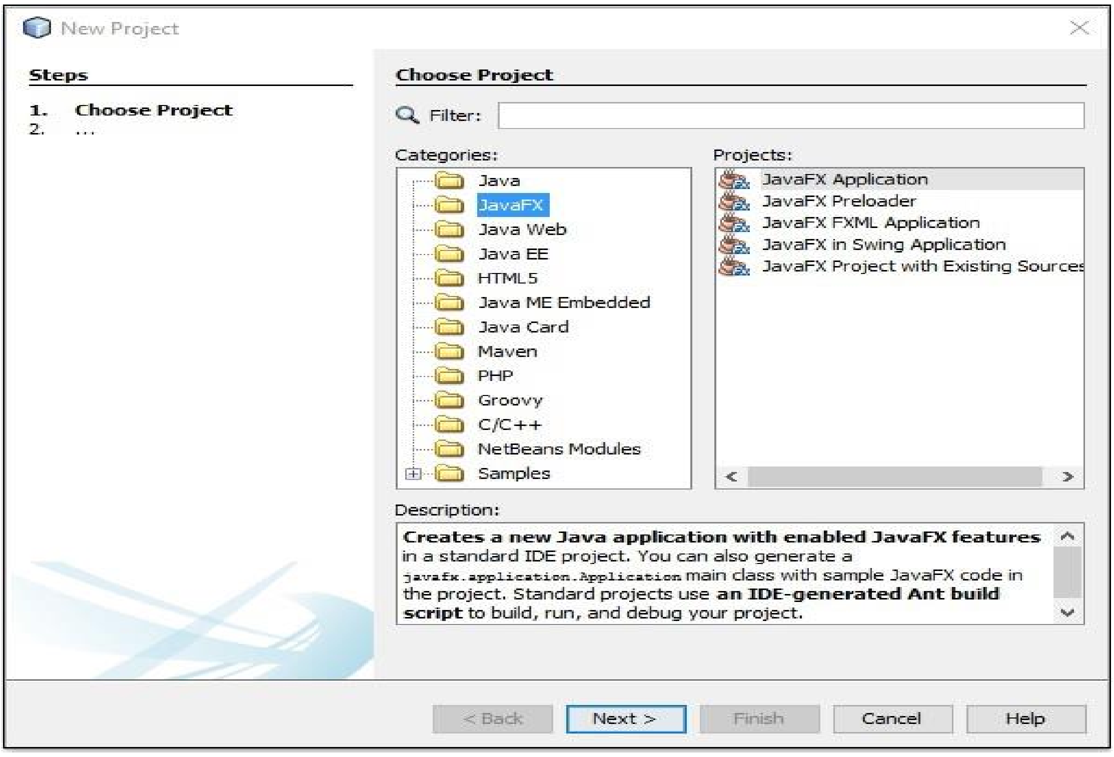
   a. **JavaFX:**
   JavaFX is a Java library used to develop Desktop applications as well as Rich Internet Applications (RIA). The applications built in JavaFX, can run on multiple platforms including Web, Mobile and Desktops.

   b. **JavaFX Features:**
   i. **FXML** − JavaFX features a language known as FXML, which is a HTML like declarative markup language.
   ii. **Built-in UI controls** − JavaFX library caters UI controls using which we can develop a full-featured application.
   iii. **Canvas-**Within the package **javafx.scene.canvas** it holds a set of classes for canvas, using which we can draw directly within an area of the JavaFX scene. JavaFX also provides classes for Printing purposes in the package **javafx.print**.
   iv. **Integrated Graphics library** − JavaFX provides classes for **2d** and **3d** graphics.

5. Laboratory Exercise

A. Steps to Begin JavaFX Project:

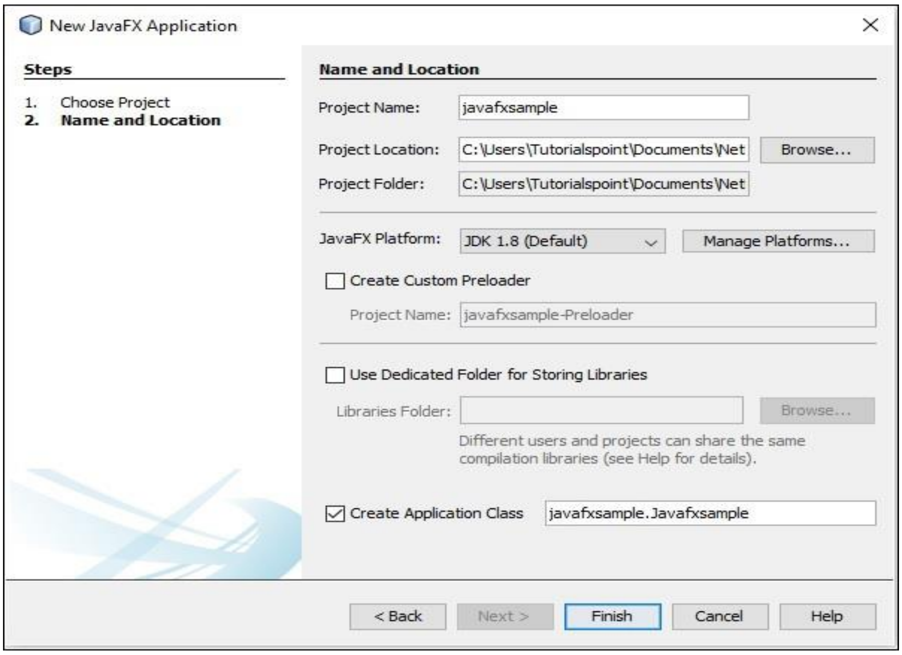




Categories: JavaFX

Projects: JavaFX Application

Description: **Creates a new Java application with enabled JavaFX features** in a standard IDE project. You can also generate a `javafx.application.Application` main class with sample JavaFX code in the project. Standard projects use **an IDE-generated Ant build script** to build, run, and debug your project.



New JavaFX Application

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name: javafxsample

Project Location: C:\Users\Tutorialspoint\Documents\Net

Project Folder: C:\Users\Tutorialspoint\Documents\Net

JavaFX Platform: JDK 1.8 (Default)

□ Create Custom Preloader

Project Name: javafxsample-Preloader

□ Use Dedicated Folder for Storing Libraries

Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

☑ Create Application Class javafxsample.Javafxsample

B. Program code with comments:

Write and execute your program code to achieve the given aim and attach it **with your own comments with neat indentation**.

C. Post-Experiments Exercise

A. Extended Theory:

1. Explain the lifecycle methods of JavaFX.

B. Results/Observations/Program output:

Present the program input/output results and comment on the same.

C. Questions/Programs:

Write Java program to draw various shapes on Canvas using JavaFX.

D. Conclusion:

1. Write what was performed in the experiment/program.
2. What is the significance of experiment/program?
3. Mention few applications of what was studied.

E. References

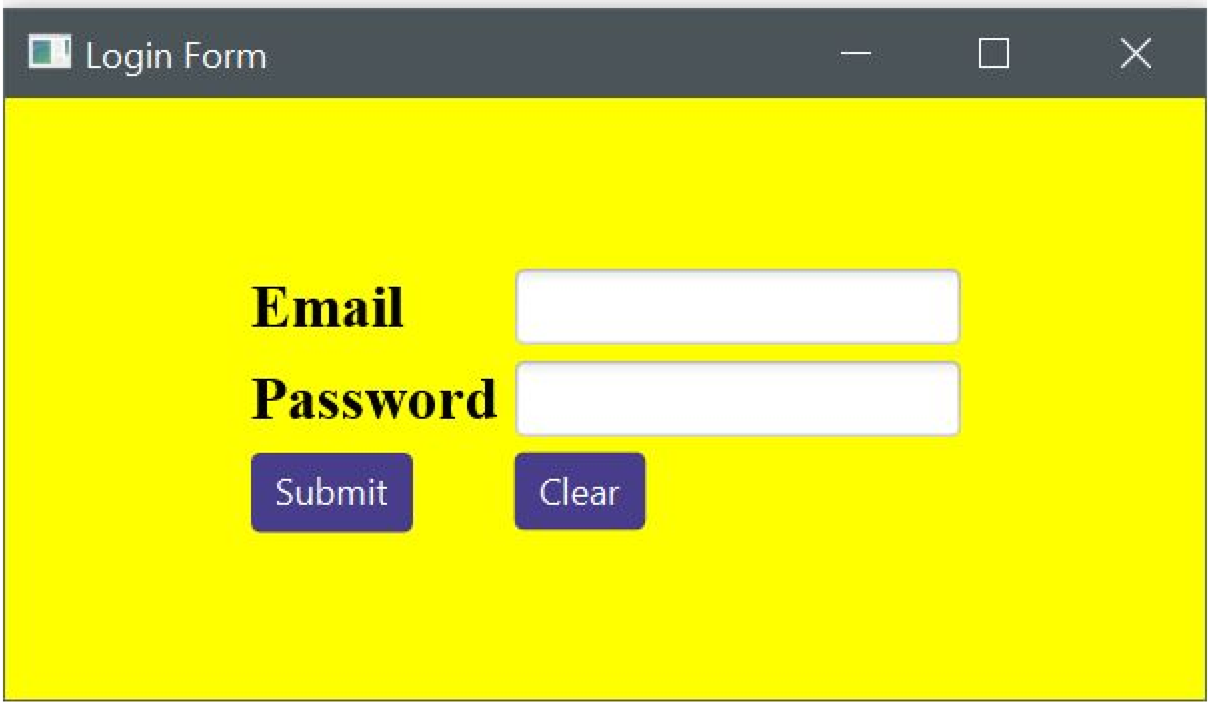
1. Java 8 Programming-Black Book,by-Dreamtech Publications.
2. www.programmingsimplified.com
3. www.javatpoint.com

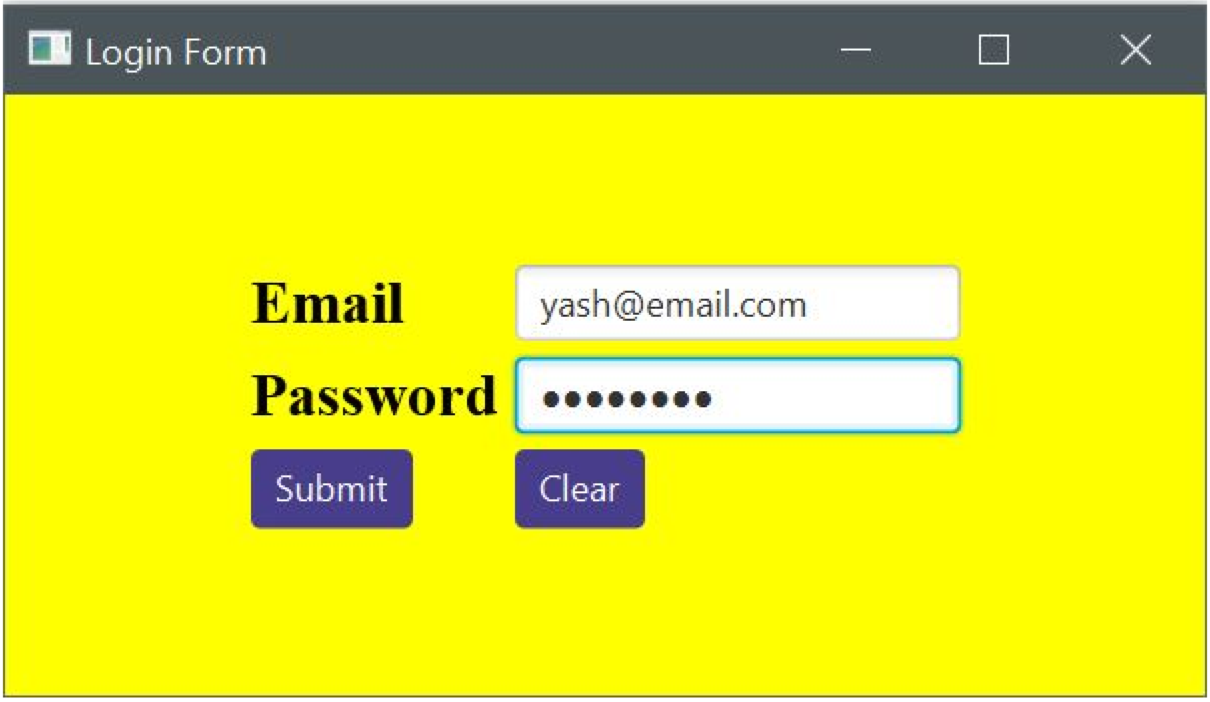
Program 1:

```
//Write a Java program to design a Login Form using JavaFX Controls.
import javafx.application.Application;
import javafx.geometry.Insets;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.PasswordField;
import javafx.scene.control.TextField;
import javafx.scene.layout.GridPane;
import javafx.scene.text.Text;
import javafx.stage.Stage;
public class Login extends Application {
       Button submit;
       Button clear;
       TextField email;
       PasswordField pass;
        @Override
   public void start(Stage s){
       Text text1 = new Text("Email");
       Text text2 = new Text("Password");
       email = new TextField();
       pass = new PasswordField();
       submit = new Button("Submit");
      submit.setOnAction(e -> {
          System.out.println("Form Submitted Successfully");

  });
      clear = new Button("Clear");
      clear.setOnAction(e -> {
        email.clear();
        pass.clear();
  });
       //Creating a Grid Pane
      GridPane gridPane = new GridPane();

      //Setting size for the pane
      gridPane.setMinSize(400, 200);
```

```java
      //Setting the padding
      gridPane.setPadding(new Insets(10, 10, 10, 10));

      //Setting the vertical and horizontal gaps between the columns
      gridPane.setVgap(5);
      gridPane.setHgap(5);

      //Setting the Grid alignment
      gridPane.setAlignment(Pos.CENTER);

      //Arranging all the nodes in the grid
      gridPane.add(text1, 0, 0);
      gridPane.add(email, 1, 0);
      gridPane.add(text2, 0, 1);
      gridPane.add(pass, 1, 1);
      gridPane.add(submit, 0, 2);
      gridPane.add(clear, 1, 2);

      //Styling nodes
      submit.setStyle("-fx-background-color: darkslateblue; -fx-text-fill: white;");
      clear.setStyle("-fx-background-color: darkslateblue; -fx-text-fill: white;");
      text1.setStyle("-fx-font: normal bold 20px 'serif' ");
      text2.setStyle("-fx-font: normal bold 20px 'serif' ");
      gridPane.setStyle("-fx-background-color: YELLOW;");
      //Creating a Scene by passing the group object, height and width
      Scene scene = new Scene(gridPane);
      //Setting the title to Stage.
      s.setTitle("Login Form");
      //Adding the scene to Stage
      s.setScene(scene);
      //Displaying the contents of the stage
      s.show();
   }
    public static void main(String[] args) {
        launch(args);
    }
}
```

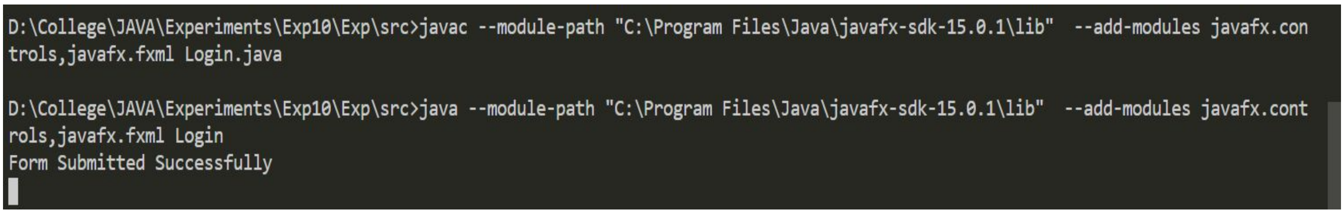
Output:





Email: yash@email.com

Password: ••••••••



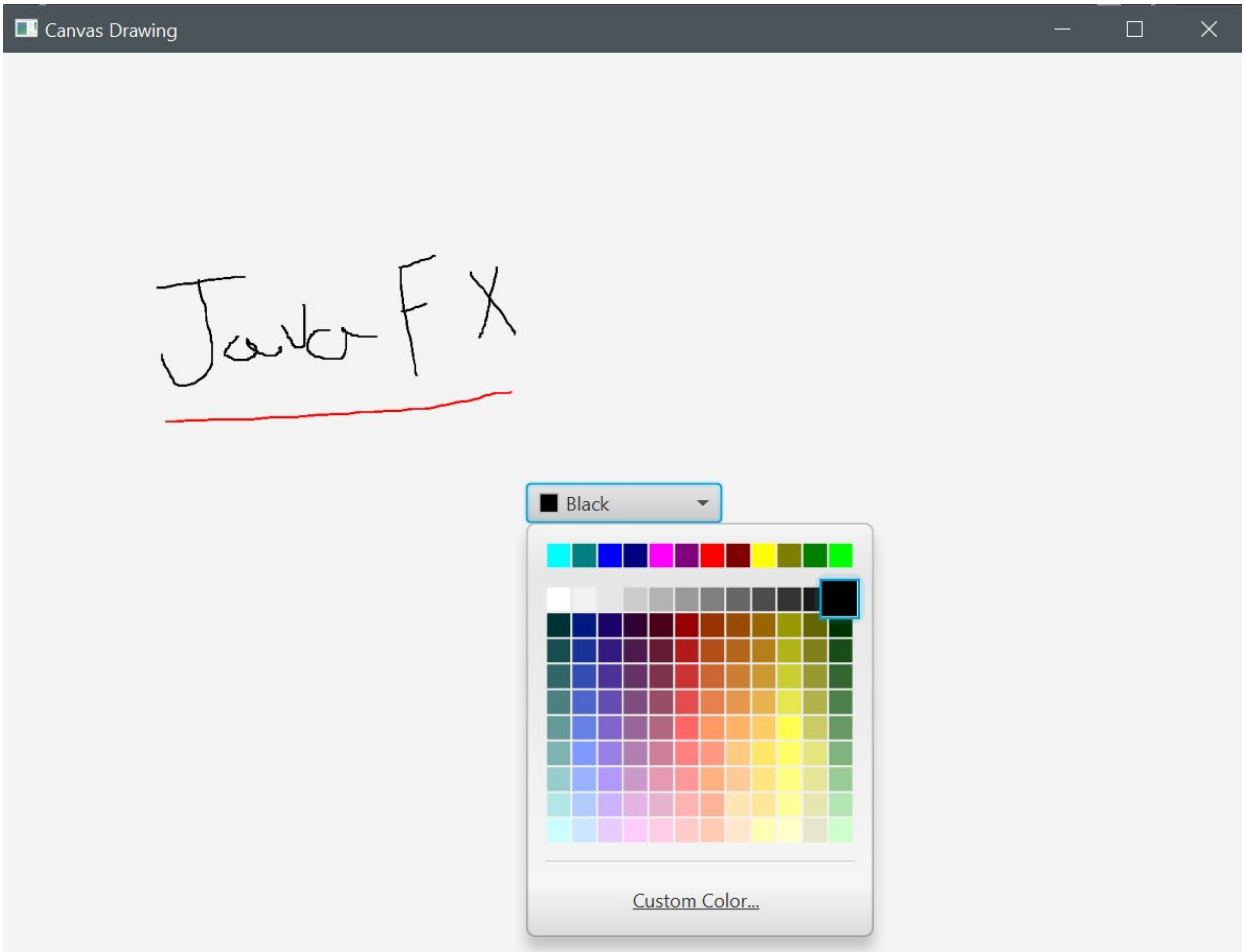
```
D:\College\JAVA\Experiments\Exp10\Exp\src>javac --module-path "C:\Program Files\Java\javafx-sdk-15.0.1\lib"  --add-modules javafx.con
trols,javafx.fxml Login.java

D:\College\JAVA\Experiments\Exp10\Exp\src>java --module-path "C:\Program Files\Java\javafx-sdk-15.0.1\lib"  --add-modules javafx.cont
rols,javafx.fxml Login
Form Submitted Successfully
```

Questions:

Question 1:

```
//Write Java program to draw various shapes on Canvas using JavaFX.
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.canvas.Canvas;
import javafx.scene.canvas.GraphicsContext;
import javafx.scene.control.ColorPicker;
import javafx.scene.layout.StackPane;
import javafx.scene.paint.Color;
import javafx.stage.Stage;


public class CanvasDrawing extends Application {

    @Override
    public void start(Stage s) {
        Canvas c=new Canvas(800,500);
        GraphicsContext gc;
        gc=c.getGraphicsContext2D();
        ColorPicker cp=new ColorPicker();
        gc.setStroke(Color.BLUE);
        gc.setLineWidth(1);
        cp.setValue(Color.BLUE);
        cp.setOnAction(e->{
            gc.setStroke(cp.getValue());
        });

        StackPane root = new StackPane();

        Scene scene = new Scene(root, 800, 500);
        scene.setOnMousePressed(e->{
            gc.beginPath();
            gc.lineTo(e.getSceneX(),e.getSceneY());
            gc.stroke();
        });
        scene.setOnMouseDragged(e->{
           gc.lineTo(e.getSceneX(),e.getSceneY());
            gc.stroke();
        });
        root.getChildren().addAll(c,cp);
        s.setTitle("Canvas Drawing");
        s.setScene(scene);
        s.show();
    }


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

}
```

Output:



Yash Mahajan SE IT B 04

6) ~~AB~~ Post - Experiment Excercise

A) Extended Theory :-

1) Explain the lifecycle of methods of JavaFx

The JavaFx application class has 3 life cycle methods:-

1) Start() - The entry point method where the JavaFx graphics code is to be written.

2) Stop () - An empty method which can be overriden, here you can write logic to stop the application.

3) init () :- An empty method which can be overriden, but you cannot create a stage or scene in this method.

In addition to these, it provides a static method named launch () to launch JavaFx application

Since the launch method is static, you need to call it from a static content, whenever a JavaFx application is launched, the action will be carried by ~~above~~ same order mention above. The last window of application ~~will~~ is ~~be~~ closed, the JavaFx application is terminated implicitly. You can turn this behaviour off by passing the boolean value false to static method set implicit Exit().

Page No :

Date :

Yash Mahajan SE IT B 04

D] Conclusion :-

→ In this experiment we have performed Java program to implement GUI using JavaFx. where

→ JavaFx is a software platform for creating & ~~dat~~ developing desktop applications aswell as rich internet applications that can run accross a wide variety of devices.

→ Javafx application code can refrence API for any Java library, for example JavaFx application can use Java API librares to access native system capabilities & ~~ca~~ connect to server using middleware application.