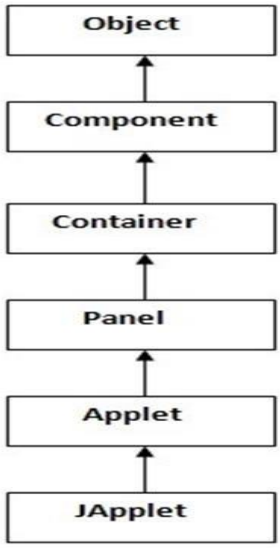
EXPERIMENT N0. 12

1. What is Applet ? What is lifecycle of applet?

Ans :

Applet is a java Program that can be embedded into a web page. It runs inside the web browser and works at client side. Applet is embedded in a HTML page using the APPLET or OBJECT tag and hosted on a web server.

Chain of classes inherited by Applet class in java.



Life Cycle of an Applet

Four methods in the Applet class gives you the framework on which you build any serious applet −

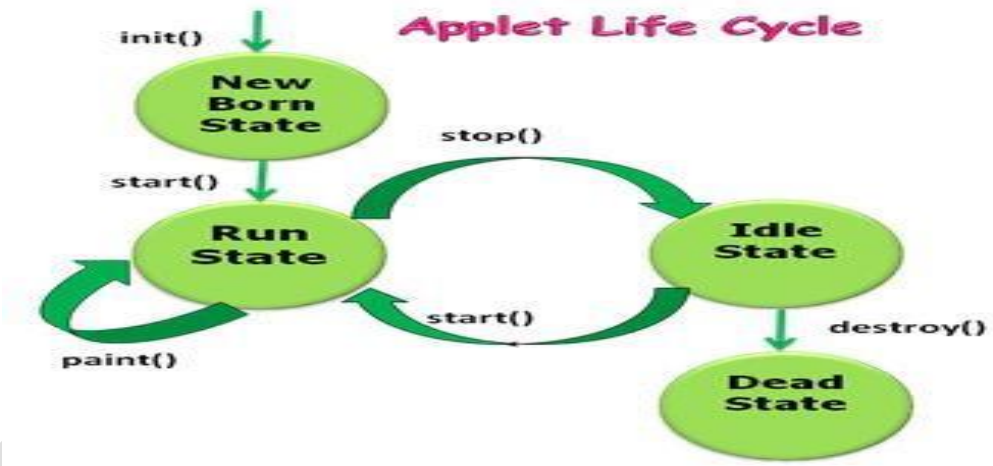
init − This method is intended for whatever initialization is needed for your applet. It is called after the param tags inside the applet tag have been processed.

start − This method is automatically called after the browser calls the init method. It is also called whenever the user returns to the page containing the applet after having gone off to other pages.

stop − This method is automatically called when the user moves off the page on which the applet sits. It can, therefore, be called repeatedly in the same applet.

destroy − This method is only called when the browser shuts down normally. Because applets are meant to live on an HTML page, you should not normally leave resources behind after a user leaves the page that contains the applet.

paint − Invoked immediately after the start() method, and also any time the applet needs to repaint itself in the browser. The paint() method is actually inherited from the java.awt.



2. **Write an example for JApplet?**

Ans:

```
// A Hello World Applet
// Save file as HelloWorld.java

import java.applet.Applet;

import java.awt.Graphics;  // HelloWorld class extends Applet

public class HelloWorld extends Applet
{
        // Overriding paint() method

        @Override
        public void paint(Graphics g)
        {

        g.drawString("Hello World", 20, 20);

        }
}
```

There are two standard ways in which you can run an applet :

i)Executing the applet within a Java-compatible web browser.

ii)Using an applet viewer, such as the standard tool, appletviewer. An applet viewer executes your applet in a window. This is generally the fastest and easiest way to test your applet .

3.**How to deploy a applet?**

Ans:

Step 1: write the java code through applet.

Step 2: compile it using command 'javac program_name.java'.
Step 3: create html file. Save with program_name.html

Add Html code in that file.

```
<html>
<head>
<title>Hello World by Your Name</title>
        </head>

<body>
<p>Hello World Applet
<applet code="HelloWorld.class" width=300 height=200>
</applet>
</body>
</html>
```

Step 4: Run the HelloWorld Applet with appletviewer.

appletviewer HelloWorld.html

Step 5: Applet output

4. **What are the new technology being available which replace applet in java?**

Ans:

Java Web Start is a better technology if you want to deploy Java applications (e.g. with a Swing GUI) to client machines. It avoids many of the restrictions of applets, while retaining the benefits of allowing you to easily deploy Java client applications over the web from a central server.

JavaFX is a much newer client side technology for Java. It has a very nice API and lots of "rich media" features. If you want to make a stylish, modern looking Java client side app it is possibly the best choice. It's still relatively new however, and not as well adopted.

JavaScript / HTML5 is probably the best choice if you want to use a different client technology (not Java). It's not as powerful as a Java client side app, but much better in terms of browser support, widget libraries etc. A great design for web applications is to use JavaScript / HTML5 for the client and a JVM language (Java, Scala, JRuby, Clojure) on the server

5. **What is javaFx ? What is use of javaFx?**

Ans:

JavaFX is a Java library used to build Rich Internet Applications. The applications written using this library can run consistently across multiple platforms. The applications developed using JavaFX can run on various devices such as Desktop Computers, Mobile Phones, TVs, Tablets, etc.

JavaFX is intended to replace swing in Java applications as a GUI framework. However, It provides more functionalities than swing. Like Swing, JavaFX also provides its own components and doesn't depend upon the operating system. It is lightweight and hardware accelerated. It supports various operating systems including Windows, Linux and Mac OS.

To develop **GUI Applications** using Java programming language, the programmers rely on libraries such as **Advanced Windowing Toolkit** and **Swing**. After the advent of JavaFX, these Java programmers can now develop GUI applications effectively with rich content.

To develop Client Side Applications with rich features, the programmers used to depend on various libraries to add features such as Media, UI controls, Web, 2D and 3D etc. **JavaFX includes all these features in a single library.**

6. **Write a code to create a registration from using JavaFx controls?**

Ans:

//Save with Registration.java

```
import javafx.application.Application;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.geometry.Insets;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.CheckBox;
 import javafx.scene.control.ChoiceBox;
```

```
import javafx.scene.control.DatePicker;
import javafx.scene.control.ListView;
import javafx.scene.control.RadioButton;
import javafx.scene.layout.GridPane;
import javafx.scene.text.Text;
import javafx.scene.control.TextField;
import javafx.scene.control.ToggleGroup;
import javafx.scene.control.ToggleButton;
import javafx.stage.Stage;

public class Registration extends Application {
@Override
public void start(Stage stage) {
Text nameLabel = new Text("Name");              //Label for name
TextField nameText = new TextField();           //Text field for name
Text dobLabel = new Text("Date of birth");      //Label for date of birth
DatePicker datePicker = new DatePicker();       //date picker to choose date
Text genderLabel = new Text("gender");          //Label for gender
ToggleGroup groupGender = new ToggleGroup();
 RadioButton maleRadio = new RadioButton("male");
maleRadio.setToggleGroup(groupGender);
RadioButton femaleRadio=newRadioButton("female");
femaleRadio.setToggleGroup(groupGender);
Text reservationLabel = new Text("Reservation");        //Label for reservation
ToggleButton Reservation = new ToggleButton();                  //Toggle button
ToggleButton yes = new ToggleButton("Yes");
ToggleButton no = new ToggleButton("No");
ToggleGroup groupReservation = new ToggleGroup();
yes.setToggleGroup(groupReservation);
no.setToggleGroup(groupReservation);
Text technologiesLabel = new Text("Technologies Known");
```

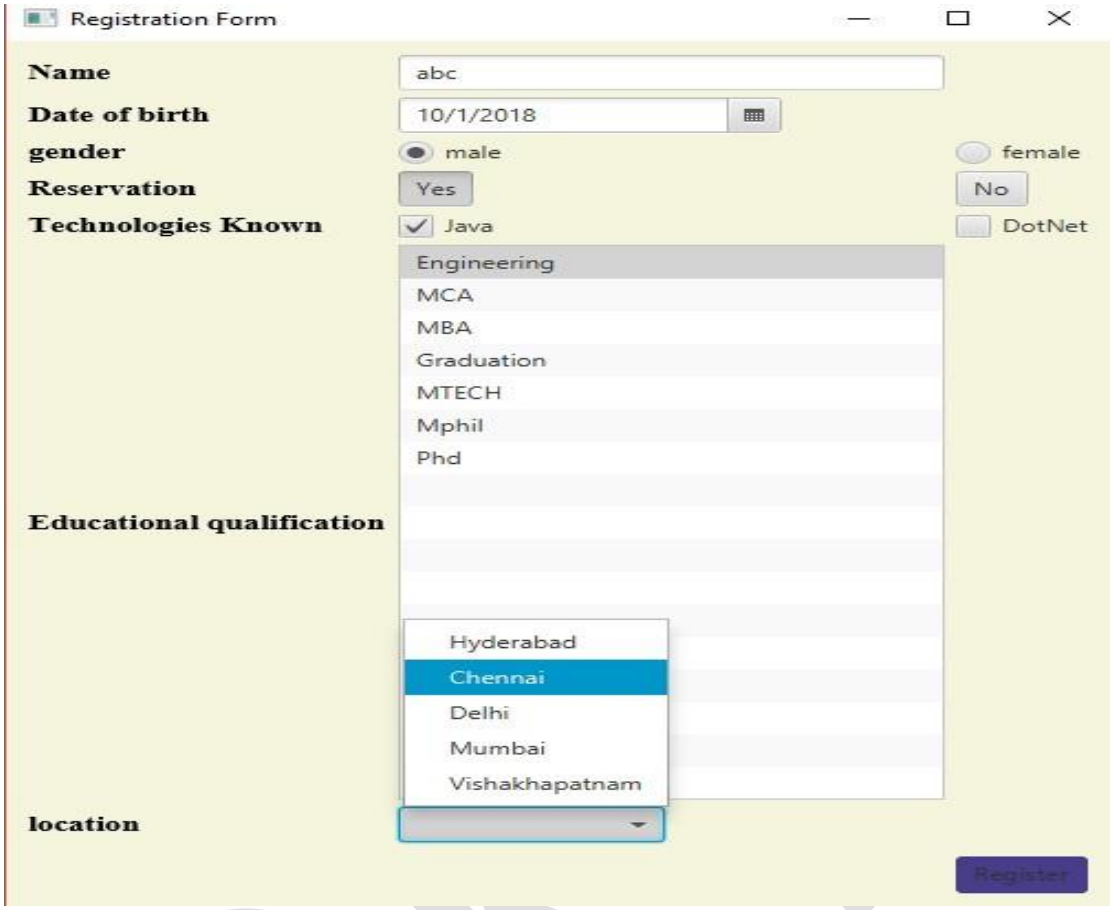
```
CheckBox javaCheckBox = new CheckBox("Java");      //check box for education
 javaCheckBox.setIndeterminate(false);

CheckBox dotnetCheckBox = new CheckBox("DotNet");
javaCheckBox.setIndeterminate(false);
Text educationLabel = new Text("Educational qualification");  //Label for edu
//list View for educational qualification
ObservableList<String> names = FXCollections.observableArrayList(
"Engineering", "MCA", "MBA", "Graduation", "MTECH", "Mphil", "Phd");
ListView<String> educationListView = new ListView<String>(names);


Text locationLabel = new Text("location");  //Label for location


ChoiceBox locationchoiceBox = new ChoiceBox();     //Choice box for location
locationchoiceBox.getItems().addAll
("Hyderabad", "Chennai", "Delhi", "Mumbai", "Vishakhapatnam");
//Label for register
Button buttonRegister = new Button("Register");

GridPane gridPane = new GridPane();    //Creating a Grid Pane
gridPane.setMinSize(500, 500);       //Setting size for the pane
gridPane.setPadding(new Insets(10, 10, 10, 10));  //Setting the padding
//Setting the vertical and horizontal gaps between the columns
gridPane.setVgap(5);
gridPane.setHgap(5);
gridPane.setAlignment(Pos.CENTER);   //Setting the Grid alignment
 gridPane.add(nameLabel, 0, 0);      //Arranging all the nodes in the grid
gridPane.add(nameText, 1, 0);
```

```
gridPane.add(dobLabel, 0, 1);
gridPane.add(datePicker, 1, 1);
gridPane.add(genderLabel, 0, 2);
gridPane.add(maleRadio, 1, 2);
gridPane.add(femaleRadio, 2, 2);
gridPane.add(reservationLabel, 0, 3);
gridPane.add(yes, 1, 3);
gridPane.add(no, 2, 3);

gridPane.add(technologiesLabel, 0, 4);
gridPane.add(javaCheckBox, 1, 4);
gridPane.add(dotnetCheckBox, 2, 4);

gridPane.add(educationLabel, 0, 5);
gridPane.add(educationListView, 1, 5);

gridPane.add(locationLabel, 0, 6);
gridPane.add(locationchoiceBox, 1, 6);

gridPane.add(buttonRegister, 2, 8);
//Styling nodes

buttonRegister.setStyle(
"-fx-background-color: darkslateblue; -fx-textfill: white;");

nameLabel.setStyle("-fx-font: normal bold 15px 'serif' ");
dobLabel.setStyle("-fx-font: normal bold 15px 'serif' ");
genderLabel.setStyle("-fx-font: normal bold 15px 'serif' ");
reservationLabel.setStyle("-fx-font: normal bold 15px 'serif' ");
technologiesLabel.setStyle("-fx-font: normal bold 15px 'serif' ");
educationLabel.setStyle("-fx-font: normal bold 15px 'serif' ");
```


```
locationLabel.setStyle("-fx-font: normal bold 15px 'serif' ");

 //Setting the back ground color
 gridPane.setStyle("-fx-background-color: BEIGE;");

 //Creating a scene object
 Scene scene = new Scene(gridPane);

 //Setting title to the Stage
 stage.setTitle("Registration Form");
 //Adding scene to the stage
 stage.setScene(scene);
 //Displaying the contents of the stage
 stage.show();
 }
 public static void main(String args[])
 {       launch(args);
 }
 }
```

To Compile and execute '.java' Program use following commands:= javac Registration.java java Registration

Output:-


Registration Form
Name
abc
Date of birth
10/1/2018
gender
male
female
Reservation
Yes
No
Technologies Known
Java
DotNet
Engineering
MCA
MBA
Graduation
MTECH
Mphil
Phd
Educational qualification
Hyderabad
Chennai
Delhi
Mumbai
Vishakhapatnam
location
Register