

[02/04, 5:55 pm] Meta AI: Here are the examples in a short and easy-to-understand format:

Basic Structure of JavaFX Program (5 marks)

- Import JavaFX packages
- Extend Application class
- Override start() method
- Create Stage and Scene
- Set title and show Stage

Example:

```

```
import javafx.application.Application;
```

```
import javafx.scene.Scene;
```

```
import javafx.scene.layout.StackPane;
```

```
import javafx.stage.Stage;
```

```
public class HelloWorld extends Application {
```

```
 @Override
```

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

```
StackPane root = new StackPane();
Scene scene = new Scene(root, 300, 250);
primaryStage.setTitle("Hello World!");
primaryStage.setScene(scene);
primaryStage.show();
}
```

```
Public static void main(String[] args) {
 Launch(args);
}
}
` ` `
```

Panes (2 marks)

- BorderPane
- StackPane
- FlowPane
- GridPane
- Hbox
- VBox

Example:

```

```
Import javafx.application.Application;
```

```
Import javafx.scene.Scene;
```

```
Import javafx.scene.layout.BorderPane;
```

```
Import javafx.stage.Stage;
```

```
Public class PaneExample extends Application {
```

```
    @Override
```

```
    Public void start(Stage primaryStage) {
```

```
        BorderPane root = new BorderPane();
```

```
        Scene scene = new Scene(root, 300, 250);
```

```
        primaryStage.setTitle("Pane Example");
```

```
        primaryStage.setScene(scene);
```

```
        primaryStage.show();
```

```
    }
```

```
    Public static void main(String[] args) {
```

```
        Launch(args);
```

```
    }
```

```
}  
` ` `
```

UI Controls and Shapes (5 marks)

- Button
- Label
- TextField
- CheckBox
- RadioButton
- Rectangle
- Circle
- Ellipse

Example:

```
` ` `
```

```
Import javafx.application.Application;  
Import javafx.scene.Scene;  
Import javafx.scene.control.Button;  
Import javafx.scene.layout.StackPane;  
Import javafx.stage.Stage;
```

```

Public class UIControlExample extends Application {

    @Override

    Public void start(Stage primaryStage) {

        Button button = new Button("Click me!");

        StackPane root = new StackPane();

        Root.getChildren().add(button);

        Scene scene = new Scene(root, 300, 250);

        primaryStage.setTitle("UI Control Example");

        primaryStage.setScene(scene);

        primaryStage.show();

    }

    Public static void main(String[] args) {

        Launch(args);

    }

}

```

Property Binding (2 marks)

- Bind properties together

Example:

```

```
Import javafx.application.Application;
```

```
Import javafx.beans.binding.Bindings;
```

```
Import javafx.scene.Scene;
```

```
Import javafx.scene.control.Label;
```

```
Import javafx.scene.control.TextField;
```

```
Import javafx.scene.layout.Vbox;
```

```
Import javafx.stage.Stage;
```

```
Public class PropertyBindingExample extends Application {
```

```
 @Override
```

```
 Public void start(Stage primaryStage) {
```

```
 TextField textField = new TextField();
```

```
 Label label = new Label();
```

```
 Label.textProperty().bind(Bindings.valueAt(textField.textProperty()));
```

```
 Vbox root = new Vbox();
```

```
 Root.getChildren().addAll(textField, label);
```

```
Scene scene = new Scene(root, 300, 250);
primaryStage.setTitle("Property Binding Example");
primaryStage.setScene(scene);
primaryStage.show();
}
```

```
Public static void main(String[] args) {
 Launch(args);
}
}
```

Color and Font Classes (2 marks)

- Color class
- Font class

Example:

```
...
```

```
Import javafx.application.Application;
Import javafx.scene.Scene;
```

```
Import javafx.scene.control.Label;
```

```
Import javafx.scene.layout.StackPane;
```

```
Import javafx.scene.paint.Color;
```

```
Import javafx.scene.text.Font;
```

```
Import javafx.stage.Stage;
```

```
Public class ColorAndFontExample extends Application {
```

```
 @Override
```

```
 Public void start(Stage primaryStage) {
```

```
 Label label = new Label("Hello, World!");
```

```
 Label.setTextFill(Color.RED);
```

```
 Label.setFont(Font.font("Arial", 24));
```

```
 StackPane root = new StackPane();
```

```
 Root.getChildren().add(label);
```

```
 Scene scene = new Scene(root, 300, 250);
```

```
 primaryStage.setTitle("Color and Font Example");
```

```
 primaryStage.setScene(scene);
```

```
 primaryStage.show();
```

```
 }
```

```
 Public static void main(String[] args) {
```



```
 Launch(args);
 }
}
` ` `
```

## Image and ImageView Classes (2 marks)

- Image class
- ImageView class

Example:

```
` ` `
```

```
Import javafx.application.Application;
Import javafx.scene.Scene;
Import javafx.scene.image.Image;
Import javafx.scene.image.ImageView;
Import javafx.scene.layout.StackPane;
Import javafx.stage.Stage;
```

```
Public class ImageExample extends Application {
 @Override
```

```

Public void start(Stage primaryStage) {
 Image image = new Image("image.jpg");
 ImageView imageView = new ImageView(image);
 StackPane root = new StackPane();
 Root.getChildren().add(imageView);
 Scene scene = new Scene(root, 300, 250);
 primaryStage.setTitle("Image Example");
 ...

```

[02/04, 5:57 pm] Meta AI: Here are the examples in a short and easy-to-understand format:

Layout Panes (5 marks)

- Use layout panes to arrange nodes
- Common layout panes: BorderPane, StackPane, FlowPane, GridPane, Hbox, VBox

Example:

```

...

Import javafx.application.Application;
Import javafx.scene.Scene;
Import javafx.scene.layout.BorderPane;

```

```
Import javafx.scene.layout.Hbox;
```

```
Import javafx.scene.control.Button;
```

```
Import javafx.stage.Stage;
```

```
Public class LayoutPaneExample extends Application {
```

```
 @Override
```

```
 Public void start(Stage primaryStage) {
```

```
 BorderPane root = new BorderPane();
```

```
 Hbox hbox = new Hbox();
```

```
 Button button1 = new Button("Button 1");
```

```
 Button button2 = new Button("Button 2");
```

```
 Hbox.getChildren().addAll(button1, button2);
```

```
 Root.setTop(hbox);
```

```
 Scene scene = new Scene(root, 300, 250);
```

```
 primaryStage.setTitle("Layout Pane Example");
```

```
 primaryStage.setScene(scene);
```

```
 primaryStage.show();
```

```
 }
```

```
Public static void main(String[] args) {
```

```
 Launch(args);
```

```
}
}
` ``
```

### Shapes (2 marks)

- Use shapes to create graphical shapes
- Common shapes: Circle, Rectangle, Ellipse

Example:

```
` ``
```

```
Import javafx.application.Application;
```

```
Import javafx.scene.Scene;
```

```
Import javafx.scene.layout.StackPane;
```

```
Import javafx.scene.shape.Circle;
```

```
Import javafx.stage.Stage;
```

```
Public class ShapeExample extends Application {
```

```
 @Override
```

```
 Public void start(Stage primaryStage) {
```

```
 Circle circle = new Circle(50);
```

```
StackPane root = new StackPane();
Root.getChildren().add(circle);
Scene scene = new Scene(root, 300, 250);
primaryStage.setTitle("Shape Example");
primaryStage.setScene(scene);
primaryStage.show();
}
```

```
Public static void main(String[] args) {
 Launch(args);
}
}
` ` `
```

### Events and Event Sources (5 marks)

- Use events to handle user interactions
- Common events: mouse clicks, key presses

Example:

` ` `

```
Import javafx.application.Application;
```

```
Import javafx.scene.Scene;
```

```
Import javafx.scene.control.Button;
```

```
Import javafx.scene.layout.StackPane;
```

```
Import javafx.stage.Stage;
```

```
Public class EventExample extends Application {
```

```
 @Override
```

```
 Public void start(Stage primaryStage) {
```

```
 Button button = new Button("Click me!");
```

```
 Button.setOnAction(event -> System.out.println("Button
clicked!"));
```

```
 StackPane root = new StackPane();
```

```
 Root.getChildren().add(button);
```

```
 Scene scene = new Scene(root, 300, 250);
```

```
 primaryStage.setTitle("Event Example");
```

```
 primaryStage.setScene(scene);
```

```
 primaryStage.show();
```

```
 }
```

```
Public static void main(String[] args) {
```

```
 Launch(args);
 }
}
` ` `
```

## Registering Handlers and Handling Events (5 marks)

- Use handlers to handle events
- Common handlers: EventHandler

Example:

```
` ` `
```

```
Import javafx.application.Application;
Import javafx.event.EventHandler;
Import javafx.scene.Scene;
Import javafx.scene.control.Button;
Import javafx.scene.layout.StackPane;
Import javafx.stage.Stage;
```

```
Public class HandlerExample extends Application {
 @Override
```

```
Public void start(Stage primaryStage) {
 Button button = new Button("Click me!");
 Button.setOnAction(new
EventHandler<javafx.event.ActionEvent>() {
 @Override
 Public void handle(javafx.event.ActionEvent event) {
 System.out.println("Button clicked!");
 }
 });
 StackPane root = new StackPane();
 Root.getChildren().add(button);
 Scene scene = new Scene(root, 300, 250);
 primaryStage.setTitle("Handler Example");
 primaryStage.setScene(scene);
 primaryStage.show();
}
```

```
Public static void main(String[] args) {
 Launch(args);
}
}
```



\\ \

### Inner Classes and Anonymous Inner Class Handlers (5 marks)

- Use inner classes to define a class within another class
- Use anonymous inner class handlers to handle events

Example:

\\ \

```
Import javafx.application.Application;
```

```
Import javafx.scene.Scene;
```

```
Import javafx.scene.control.Button;
```

```
Import javafx.scene.layout.StackPane;
```

```
Import javafx.stage.Stage;
```

```
Public class InnerClassExample extends Application {
```

```
 @Override
```

```
 Public void start(Stage primaryStage) {
```

```
 Button button = new Button("Click me!");
```

```
 Button.setOnAction(new ButtonHandler());
```

```
 StackPane root = new StackPane();
```

```
Root.getChildren().add(button);
Scene scene = new Scene(root, 300, 250);
primaryStage.setTitle("Inner Class Example");
primaryStage.setScene(scene);
primaryStage.show();
}
```

```
Private class ButtonHandler implements
EventHandler<javafx.event.ActionEvent> {
 @Override
 Public void handle(javafx.event.ActionEvent event) {
 System.out.println("Button clicked!");
 }
}
```

```
Public static void main(String[] args) {
 Launch(args);
}
}
```

...

## Mouse and Key Events (5 marks)

- Use mouse events to handle mouse interactions
- Use key events to handle keyboard interactions

Example:

```
```
```

```
import javafx.application.Application;
```

```
import javafx.scene.Scene;
```

```
import javafx.scene.layout.Stack
```

```
```
```

[02/04, 5:58 pm] Meta AI: Here are the examples in a short and easy-to-understand format:

## Mouse Events (5 marks)

- Handle mouse interactions: clicking, hovering, dragging
- Use `setOnMouseClicked`, `setOnMouseEntered`, `setOnMouseExited`

Example:

```
```
```

```
Import javafx.application.Application;
```

```
Import javafx.scene.Scene;
```

```
Import javafx.scene.layout.StackPane;
```

```
Import javafx.scene.shape.Circle;
```

```
Import javafx.stage.Stage;
```

```
Public class MouseEventExample extends Application {
```

```
    @Override
```

```
    Public void start(Stage primaryStage) {
```

```
        Circle circle = new Circle(50);
```

```
        Circle.setOnMouseClicked(event -> System.out.println("Mouse  
clicked!"));
```

```
        StackPane root = new StackPane();
```

```
        Root.getChildren().add(circle);
```

```
        Scene scene = new Scene(root, 300, 250);
```

```
        primaryStage.setTitle("Mouse Event Example");
```

```
        primaryStage.setScene(scene);
```

```
        primaryStage.show();
```

```
    }
```

```
Public static void main(String[] args) {
```

```
        Launch(args);  
    }  
}  
` ` `
```

Key Events (5 marks)

- Handle keyboard interactions: pressing, releasing keys
- Use `setOnKeyPressed`, `setOnKeyReleased`, `setOnKeyTyped`

Example:

```
` ` `
```

```
Import javafx.application.Application;  
Import javafx.scene.Scene;  
Import javafx.scene.layout.StackPane;  
Import javafx.scene.shape.Circle;  
Import javafx.stage.Stage;
```

```
Public class KeyEventExample extends Application {  
    @Override  
    Public void start(Stage primaryStage) {
```

```

    Circle circle = new Circle(50);

    Circle.setOnKeyPressed(event -> System.out.println("Key
pressed: " + event.getCode()));

    StackPane root = new StackPane();

    Root.getChildren().add(circle);

    Scene scene = new Scene(root, 300, 250);

    primaryStage.setTitle("Key Event Example");

    primaryStage.setScene(scene);

    primaryStage.show();
}

Public static void main(String[] args) {

    Launch(args);

}

...

```

Listeners for Observable Objects (5 marks)

- Observe changes to observable objects
- Use addListener to listen for changes

Example:

```

```
Import javafx.application.Application;
```

```
Import javafx.beans.value.ChangeListener;
```

```
Import javafx.beans.value.ObservableValue;
```

```
Import javafx.scene.Scene;
```

```
Import javafx.scene.control.CheckBox;
```

```
Import javafx.scene.layout.StackPane;
```

```
Import javafx.stage.Stage;
```

```
Public class ListenerExample extends Application {
```

```
 @Override
```

```
 Public void start(Stage primaryStage) {
```

```
 CheckBox checkBox = new CheckBox("Check me!");
```

```
 checkBox.selectedProperty().addListener(new
ChangeListener<Boolean>() {
```

```
 @Override
```

```
 Public void changed(ObservableValue<? Extends Boolean>
observable, Boolean oldValue, Boolean newValue) {
```

```
 System.out.println("CheckBox selected: " + newValue);
```

```
 }
```

```

});

StackPane root = new StackPane();

Root.getChildren().add(checkBox);

Scene scene = new Scene(root, 300, 250);

primaryStage.setTitle("Listener Example");

primaryStage.setScene(scene);

primaryStage.show();

}

Public static void main(String[] args) {

 Launch(args);

}

}

...

```

Animation (5 marks)

- Create dynamic visual effects
- Use Timeline, KeyFrame to animate

Example:



...

```
Import javafx.animation.Animation;
```

```
Import javafx.animation.KeyFrame;
```

```
Import javafx.animation.Timeline;
```

```
Import javafx.application.Application;
```

```
Import javafx.scene.Scene;
```

```
Import javafx.scene.layout.StackPane;
```

```
Import javafx.scene.shape.Circle;
```

```
Import javafx.stage.Stage;
```

```
Import javafx.util.Duration;
```

```
Public class AnimationExample extends Application {
```

```
 @Override
```

```
 Public void start(Stage primaryStage) {
```

```
 Circle circle = new Circle(50);
```

```
 Timeline timeline = new Timeline();
```

```
 Timeline.setCycleCount(Animation.INDEFINITE);
```

```
 KeyFrame keyFrame = new KeyFrame(Duration.millis(1000),
event -> circle.setRadius(circle.getRadius() + 10));
```

```
 Timeline.getKeyFrames().add(keyFrame);
```

```
 Timeline.play();
```

```
StackPane root = new StackPane();
Root.getChildren().add(circle);
Scene scene = new Scene(root, 300, 250);
primaryStage.setTitle("Animation Example");
primaryStage.setScene(scene);
primaryStage.show();
}
```

```
Public static void main(String[] args) {
 Launch(args);
}
}
` ``
```