

APP WEEK 9

**Adya Singh
RA2211003010181**

Q1

```
import javax.swing.*;  
import java.awt.*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
  
// Create a custom class that extends JFrame (inheritance)  
class My SwingApp extends JFrame {  
  
    // Constructor for the custom class  
    public My SwingApp() {  
        // Set the title for the window  
        super("Swing Inheritance Example");  
  
        // Create a button  
        JButton button = new JButton("Click Me");  
  
        // Add an ActionListener to the button  
        button.addActionListener(new ActionListener() {  
            @Override  
            public void actionPerformed(ActionEvent e) {  
                // Show a dialog box with a message when the button is clicked  
                JOptionPane.showMessageDialog(null, "Hello, Swing World!");  
            }  
        });  
  
        // Set the layout manager for the frame  
        setLayout(new FlowLayout());  
  
        // Add the button to the frame  
        add(button);  
  
        // Set the default close operation  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
  
        // Set the size of the frame  
        setSize(300, 150);  
  
        // Center the frame on the screen  
        setLocationRelativeTo(null);  
  
        // Make the frame visible  
        setVisible(true);  
    }  
  
    public static void main(String[] args) {  
        // Create an instance of the custom class  
        SwingUtilities.invokeLater(new Runnable() {  
            @Override  
            public void run() {  
                new My SwingApp();  
            }  
        });  
    }  
  
}  
  
public class SwingInheritanceExample {  
    public static void main(String[] args) {  
        // This main method is not used because the My SwingApp class has its own main method.  
    }  
}
```

```
...Program finished with exit code 0  
Press ENTER to exit console.
```

Q2

```
import javax.swing.*;  
import java.awt.*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
  
public class SwingActionListenerExample {  
    public static void main(String[] args) {  
        // Create the main frame  
        JFrame frame = new JFrame("Swing ActionListener  
Example");  
  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        frame.setSize(300, 150);  
  
        // Create a button  
        JButton button = new JButton("Click Me");  
  
        // Create a label to display the click count  
        JLabel label = new JLabel("Click count: 0");  
  
        // Create an ActionListener for the button  
        ActionListener actionListener = new ActionListener() {  
            private int clickCount = 0;  
  
            @Override  
            public void actionPerformed(ActionEvent e) {  
                clickCount++;  
                label.setText("Click count: " + clickCount);  
            }  
        };  
  
        // Add the ActionListener to the button  
        button.addActionListener(actionListener);  
  
        // Create a panel to hold the button and label  
        JPanel panel = new JPanel();  
        panel.setLayout(new FlowLayout());  
        panel.add(button);  
        panel.add(label);  
  
        // Add the panel to the frame  
        frame.add(panel);  
  
        // Center the frame on the screen  
        frame.setLocationRelativeTo(null);  
  
        // Make the frame visible  
        frame.setVisible(true);  
    }  
}
```

```
Exception in thread "main" java.awt.AWTError: Can't connect to X11 window server using ':1' as the value of the DISPLAY variable
.
at java.desktop/sun.awt.X11GraphicsEnvironment.initDisplay(Native Method)
at java.desktop/sun.awt.X11GraphicsEnvironment$1.run(X11GraphicsEnvironment.java:102)
at java.base/java.security.AccessController.doPrivileged(Native Method)
at java.desktop/sun.awt.X11GraphicsEnvironment.<clinit>(X11GraphicsEnvironment.java:61)
at java.base/java.lang.Class.forName0(Native Method)
at java.base/java.lang.Class.forName(Class.java:315)
at java.desktop/java.awt.GraphicsEnvironment$LocalGE.createGE(GraphicsEnvironment.java:101)
at java.desktop/java.awt.GraphicsEnvironment$LocalGE.<clinit>(GraphicsEnvironment.java:83)
at java.desktop/java.awt.GraphicsEnvironment.getLocalGraphicsEnvironment(GraphicsEnvironment.java:129)
at java.desktop/java.awt.Window.initGC(Window.java:487)
at java.desktop/java.awt.Window.init(Window.java:507)
at java.desktop/java.awt.Window.<init>(Window.java:549)
at java.desktop/java.awt.Frame.<init>(Frame.java:423)
at java.desktop/javax.swing.JFrame.<init>(JFrame.java:224)
at SwingActionListenerExample.main(SwingActionListenerExample.java:9)

...Program finished with exit code 1
Press ENTER to exit console.[]
```

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class MenuExample {
    public static void main(String[] args) {
        // Create the main frame
        JFrame frame = new JFrame("Menu Example");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(400, 300);

        // Create a text area to display menu item selections
        JTextArea textArea = new JTextArea();
        textArea.setEditable(false);
        JScrollPane scrollPane = new JScrollPane(textArea);

        // Create a menu bar
        JMenuBar menuBar = new JMenuBar();

        // Create menus
        JMenu fileMenu = new JMenu("File");
        JMenu editMenu = new JMenu("Edit");

        // Create menu items for the "File" menu
        JMenuItem openItem = new JMenuItem("Open");
        JMenuItem saveItem = new JMenuItem("Save");
        JMenuItem exitItem = new JMenuItem("Exit");

        // Create menu items for the "Edit" menu
        JMenuItem cutItem = new JMenuItem("Cut");
        JMenuItem copyItem = new JMenuItem("Copy");
        JMenuItem pasteItem = new JMenuItem("Paste");

        // Add action listeners to menu items
        openItem.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                textArea.append("File -> Open selected\n");
            }
        });

        saveItem.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                textArea.append("File -> Save selected\n");
            }
        });

        exitItem.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                System.exit(0);
            }
        });

        cutItem.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                textArea.append("Edit -> Cut selected\n");
            }
        });

        copyItem.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                textArea.append("Edit -> Copy selected\n");
            }
        });

        pasteItem.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                textArea.append("Edit -> Paste selected\n");
            }
        });

        // Add menu items to the "File" menu
        fileMenu.add(openItem);
        fileMenu.add(saveItem);
        fileMenu.addSeparator(); // Add a separator line
        fileMenu.add(exitItem);

        // Add menu items to the "Edit" menu
        editMenu.add(cutItem);
        editMenu.add(copyItem);
        editMenu.add(pasteItem);

        // Add menus to the menu bar
        menuBar.add(fileMenu);
        menuBar.add(editMenu);

        // Set the menu bar for the frame
        frame.setJMenuBar(menuBar);

        // Create a panel to hold the text area
        JPanel panel = new JPanel(new BorderLayout());
        panel.add(scrollPane, BorderLayout.CENTER);

        // Add the panel to the frame
        frame.add(panel);

        // Center the frame on the screen
        frame.setLocationRelativeTo(null);

        // Make the frame visible
        frame.setVisible(true);
    }
}
```

```
Exception in thread "main" java.awt.AWTError: Can't connect to X11 window server using ':1' as the value of the DISPLAY variable
.
   at java.desktop/sun.awt.X11GraphicsEnvironment.initDisplay(Native Method)
   at java.desktop/sun.awt.X11GraphicsEnvironment$1.run(X11GraphicsEnvironment.java:102)
   at java.base/java.security.AccessController.doPrivileged(Native Method)
   at java.desktop/sun.awt.X11GraphicsEnvironment.<clinit>(X11GraphicsEnvironment.java:61)
   at java.base/java.lang.Class.forName0(Native Method)
   at java.base/java.lang.Class.forName(Class.java:315)
   at java.desktop/java.awt.GraphicsEnvironment$LocalGE.createGE(GraphicsEnvironment.java:101)
   at java.desktop/java.awt.GraphicsEnvironment$LocalGE.<clinit>(GraphicsEnvironment.java:83)
   at java.desktop/java.awt.GraphicsEnvironment.getLocalGraphicsEnvironment(GraphicsEnvironment.java:129)
   at java.desktop/java.awt.Window.initGC(Window.java:487)
   at java.desktop/java.awt.Window.init(Window.java:507)
   at java.desktop/java.awt.Window.<init>(Window.java:549)
   at java.desktop/java.awt.Frame.<init>(Frame.java:423)
   at java.desktop/javax.swing.JFrame.<init>(JFrame.java:224)
   at MenuExample.main(MenuExample.java:9)

...Program finished with exit code 1
Press ENTER to exit console.
```

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class StudentRegistrationForm {
    public static void main(String[] args) {
        // Create the main frame
        JFrame frame = new JFrame("Student Registration Form");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(400, 300);

        // Create a panel to hold form components
        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(6, 2, 10, 10)); // Rows, columns, horizontal gap, vertical gap

        // Create labels and text fields for form elements
        JLabel nameLabel = new JLabel("Name:");
        JTextField nameField = new JTextField();

        JLabel rollNoLabel = new JLabel("Roll Number:");
        JTextField rollNoField = new JTextField();

        JLabel genderLabel = new JLabel("Gender:");
        JRadioButton maleRadioButton = new JRadioButton("Male");
        JRadioButton femaleRadioButton = new JRadioButton("Female");
        ButtonGroup genderGroup = new ButtonGroup();
        genderGroup.add(maleRadioButton);
        genderGroup.add(femaleRadioButton);

        JLabel courseLabel = new JLabel("Course:");
        String[] courses = {"Select Course", "Computer Science", "Electrical Engineering", "Mechanical Engineering"};
        JComboBox<String> courseComboBox = new JComboBox(courses);

        JLabel cityLabel = new JLabel("City:");
        JTextField cityField = new JTextField();

        // Create a button to submit the form
        JButton submitButton = new JButton("Submit");

        // Add form components to the panel
        panel.add(nameLabel);
        panel.add(nameField);
        panel.add(rollNoLabel);
        panel.add(rollNoField);
        panel.add(genderLabel);
        panel.add(maleRadioButton);
        panel.add(new JLabel()); // Empty label for spacing
        panel.add(femaleRadioButton);
        panel.add(courseLabel);
        panel.add(courseComboBox);
        panel.add(cityLabel);
        panel.add(cityField);

        // Add an action listener to the submit button
        submitButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                // Retrieve and display form data when the submit button is clicked
                String name = nameField.getText();
                String rollNo = rollNoField.getText();
                String gender = maleRadioButton.isSelected() ? "Male" : "Female";
                String course = (String) courseComboBox.getSelectedItem();
                String city = cityField.getText();

                String message = "Name: " + name + "\n" +
                                "Roll Number: " + rollNo + "\n" +
                                "Gender: " + gender + "\n" +
                                "Course: " + course + "\n" +
                                "City: " + city;

                JOptionPane.showMessageDialog(frame, message, "Registration Details", JOptionPane.INFORMATION_MESSAGE);
            }
        });

        // Add the panel and submit button to the frame
        frame.add(panel, BorderLayout.CENTER);
        frame.add(submitButton, BorderLayout.SOUTH);

        // Center the frame on the screen
        frame.setLocationRelativeTo(null);

        // Make the frame visible
        frame.setVisible(true);
    }
}
```

```
Exception in thread "main" java.awt.AWTError: Can't connect to X11 window server using ':1' as the value of the DISPLAY variable
.
at java.desktop/sun.awt.X11GraphicsEnvironment.initDisplay(Native Method)
at java.desktop/sun.awt.X11GraphicsEnvironment$1.run(X11GraphicsEnvironment.java:102)
at java.base/java.security.AccessController.doPrivileged(Native Method)
at java.desktop/sun.awt.X11GraphicsEnvironment.<clinit>(X11GraphicsEnvironment.java:61)
at java.base/java.lang.Class.forName0(Native Method)
at java.base/java.lang.Class.forName(Class.java:315)
at java.desktop/java.awt.GraphicsEnvironment$LocalGE.createGE(GraphicsEnvironment.java:101)
at java.desktop/java.awt.GraphicsEnvironment$LocalGE.<clinit>(GraphicsEnvironment.java:83)
at java.desktop/java.awt.GraphicsEnvironment.getLocalGraphicsEnvironment(GraphicsEnvironment.java:129)
at java.desktop/java.awt.Window.initGC(Window.java:487)
at java.desktop/java.awt.Window.init(Window.java:507)
at java.desktop/java.awt.Window.<init>(Window.java:549)
at java.desktop/java.awt.Frame.<init>(Frame.java:423)
at java.desktop/javax.swing.JFrame.<init>(JFrame.java:224)
at StudentRegistrationForm.main(StudentRegistrationForm.java:9)

...Program finished with exit code 1
Press ENTER to exit console.
```

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class EmploymentRegistrationForm {
    public static void main(String[] args) {
        // Create the main frame
        JFrame frame = new JFrame("Employment Registration Form");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(400, 300);

        // Create a panel to hold form components
        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(7, 2, 10, 10)); // Rows, columns, horizontal gap, vertical gap

        // Create labels and text fields for form elements
        JLabel nameLabel = new JLabel("Name:");
        JTextField nameField = new JTextField();

        JLabel genderLabel = new JLabel("Gender:");
        JRadioButton maleRadioButton = new JRadioButton("Male");
        JRadioButton femaleRadioButton = new JRadioButton("Female");
        ButtonGroup genderGroup = new ButtonGroup();
        genderGroup.add(maleRadioButton);
        genderGroup.add(femaleRadioButton);

        JLabel emailLabel = new JLabel("Email:");
        JTextField emailField = new JTextField();

        JLabel phoneLabel = new JLabel("Phone:");
        JTextField phoneField = new JTextField();

        JLabel positionLabel = new JLabel("Position:");
        String[] positions = {"Select Position", "Software Engineer", "Data Analyst", "Project Manager"};
        JComboBox<String> positionComboBox = new JComboBox(positions);

        JLabel experienceLabel = new JLabel("Years of Experience:");
        JTextField experienceField = new JTextField();

        // Create a button to submit the form
        JButton submitButton = new JButton("Submit");

        // Add form components to the panel
        panel.add(nameLabel);
        panel.add(nameField);
        panel.add(genderLabel);
        panel.add(maleRadioButton);
        panel.add(new JLabel()); // Empty label for spacing
        panel.add(femaleRadioButton);
        panel.add(emailLabel);
        panel.add(emailField);
        panel.add(phoneLabel);
        panel.add(phoneField);
        panel.add(positionLabel);
        panel.add(positionComboBox);
        panel.add(experienceLabel);
        panel.add(experienceField);

        // Add an action listener to the submit button
        submitButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                // Retrieve and display form data when the submit button is clicked
                String name = nameField.getText();
                String gender = maleRadioButton.isSelected() ? "Male" : "Female";
                String email = emailField.getText();
                String phone = phoneField.getText();
                String position = (String) positionComboBox.getSelectedItem();
                String experience = experienceField.getText();

                String message = "Name: " + name + "\n" +
                                "Gender: " + gender + "\n" +
                                "Email: " + email + "\n" +
                                "Phone: " + phone + "\n" +
                                "Position: " + position + "\n" +
                                "Experience: " + experience + " years";

                JOptionPane.showMessageDialog(frame, message, "Registration Details", JOptionPane.INFORMATION_MESSAGE);
            }
        });

        // Add the panel and submit button to the frame
        frame.add(panel, BorderLayout.CENTER);
        frame.add(submitButton, BorderLayout.SOUTH);

        // Center the frame on the screen
        frame.setLocationRelativeTo(null);

        // Make the frame visible
        frame.setVisible(true);
    }
}
```

```
exception in thread "main" java.awt.AWTError: Can't connect to X11 window server using ':1' as the value of the DISPLAY variable  
at java.desktop/sun.awt.X11GraphicsEnvironment.initDisplay(Native Method)  
at java.desktop/sun.awt.X11GraphicsEnvironment$1.run(X11GraphicsEnvironment.java:102)  
at java.base/java.security.AccessController.doPrivileged(Native Method)  
at java.desktop/sun.awt.X11GraphicsEnvironment.<clinit>(X11GraphicsEnvironment.java:61)  
at java.base/java.lang.Class.forName0(Native Method)  
at java.base/java.lang.Class.forName(Class.java:315)  
at java.desktop/java.awt.GraphicsEnvironment$LocalGE.createGE(GraphicsEnvironment.java:101)  
at java.desktop/java.awt.GraphicsEnvironment$LocalGE.<clinit>(GraphicsEnvironment.java:83)  
at java.desktop/java.awt.GraphicsEnvironment.getLocalGraphicsEnvironment(GraphicsEnvironment.java:129)  
at java.desktop/java.awt.Window.initGC(Window.java:487)  
at java.desktop/java.awt.Window.init(Window.java:507)  
at java.desktop/java.awt.Window.<init>(Window.java:549)  
at java.desktop/java.awt.Frame.<init>(Frame.java:423)  
at java.desktop/javax.swing.JFrame.<init>(JFrame.java:224)  
at EmploymentRegistrationForm.main(EmploymentRegistrationForm.java:9)  
  
..Program finished with exit code 1  
Press ENTER to exit console.
```

```
import java.applet.Applet;
import java.awt.Color;
import java.awt.Graphics;

public class ShapeDrawingApplet extends Applet {

    @Override
    public void paint(Graphics g) {
        // Draw an oval
        g.setColor(Color.RED);
        g.fillOval(50, 50, 100, 100);

        // Draw a rectangle
        g.setColor(Color.GREEN);
        g.fillRect(200, 50, 100, 100);

        // Draw a line
        g.setColor(Color.BLUE);
        g.drawLine(50, 200, 250, 200);
    }
}
```

```
Error: Main method not found in class ShapeDrawingApplet, please define the main method as:  
    public static void main(String[] args)  
or a JavaFX application class must extend javafx.application.Application  
  
...Program finished with exit code 1  
Press ENTER to exit console.
```

```
import java.applet.Applet;
import java.awt.Color;
import java.awt.Graphics;

public class ChessboardApplet extends Applet {

    @Override
    public void paint(Graphics g) {
        int squareSize = 50; // Size of each square (width and height)
        int numRows = 8; // Number of rows on the chessboard
        int numCols = 8; // Number of columns on the chessboard

        // Loop through rows
        for (int row = 0; row < numRows; row++) {
            // Loop through columns
            for (int col = 0; col < numCols; col++) {
                // Calculate the x and y coordinates for the current square
                int x = col * squareSize;
                int y = row * squareSize;

                // Set the color for the square based on row and column (alternating black and white)
                if ((row + col) % 2 == 0) {
                    g.setColor(Color.WHITE);
                } else {
                    g.setColor(Color.BLACK);
                }

                // Fill the square with the selected color
                g.fillRect(x, y, squareSize, squareSize);
            }
        }
    }
}
```

```
Error: Main method not found in class ChessboardApplet, please define the main method as:  
      public static void main(String[] args)  
or a JavaFX application class must extend javafx.application.Application  
  
...Program finished with exit code 1  
Press ENTER to exit console.
```

```
import java.applet.Applet;
import java.awt.*;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;

public class MouseEventApplet extends Applet {
    private String eventName = "Mouse Event";

    public void init() {
        // Add a MouseAdapter to the Applet
        addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
                eventName = "Mouse Clicked";
                repaint();
            }

            @Override
            public void mousePressed(MouseEvent e) {
                eventName = "Mouse Pressed";
                repaint();
            }

            @Override
            public void mouseReleased(MouseEvent e) {
                eventName = "Mouse Released";
                repaint();
            }

            @Override
            public void mouseEntered(MouseEvent e) {
                eventName = "Mouse Entered";
                repaint();
            }

            @Override
            public void mouseExited(MouseEvent e) {
                eventName = "Mouse Exited";
                repaint();
            }
        });
    }

    public void paint(Graphics g) {
        // Clear the applet's content
        g.clearRect(0, 0, getWidth(), getHeight());

        // Get the center coordinates of the applet
        int centerX = getWidth() / 2;
        int centerY = getHeight() / 2;

        // Set the font and color for displaying the event name
        g.setFont(new Font("Arial", Font.BOLD, 16));
        g.setColor(Color.BLUE);

        // Display the event name at the center of the window
        g.drawString(eventName, centerX - 50, centerY);
    }
}
```

```
Error: Main method not found in class MouseEventApplet, please define the main method as:  
    public static void main(String[] args)  
or a JavaFX application class must extend javafx.application.Application  
  
...Program finished with exit code 1  
Press ENTER to exit console.
```

```
class Student {  
    private String name;  
    private int rollNo;  
  
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name){  
        this.name = name;  
    }  
  
    public int getRollNo() {  
        return rollNo;  
    }  
  
    public void setRollNo(int rollNo){  
        this.rollNo = rollNo;  
    }  
  
}  
  
// View (StudentView class)  
class StudentView {  
    public void printStudentDetails(String studentName, int studentRollNo){  
        System.out.println("Student:");  
        System.out.println("Name: " + studentName);  
        System.out.println("Roll No: " + studentRollNo);  
    }  
  
}  
  
// Controller (StudentController class)  
class StudentController {  
    private Student model;  
    private StudentView view;  
  
    public StudentController(Student model, StudentView view){  
        this.model = model;  
        this.view = view;  
    }  
  
    public void setStudentName(String name) {  
        model.setName(name);  
    }  
  
    public String getStudentName() {  
        return model.getName();  
    }  
  
    public void setStudentRollNo(int rollNo) {  
        model.setRollNo(rollNo);  
    }  
  
    public int getStudentRollNo() {  
        return model.getRollNo();  
    }  
  
    public void updateView() {  
        view.printStudentDetails(model.getName(), model.getRollNo());  
    }  
  
}  
  
public class MVCPatternExample {  
    public static void main(String[] args) {  
        // Create a Student model  
        Student student = new Student();  
        student.setName("John");  
        student.setRollNo(101);  
  
        // Create a StudentView  
        StudentView studentView = new StudentView();  
  
        // Create a StudentController with the model and view  
        StudentController studentController = new StudentController(student, studentView);  
  
        // Update the model data through the controller  
        studentController.setStudentName("Alice");  
        studentController.setStudentRollNo(102);  
  
        // Display the updated data through the view  
        studentController.updateView();  
    }  
}
```

```
Student:  
Name: Alice  
Roll No: 102
```

```
...Program finished with exit code 0  
Press ENTER to exit console.
```

```
class Employee {
    private String name;
    private int employeeId;
    private double salary;

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getEmployeeId() {
        return employeeId;
    }

    public void setEmployeeId(int employeeId) {
        this.employeeId = employeeId;
    }

    public double getSalary() {
        return salary;
    }

    public void setSalary(double salary) {
        this.salary = salary;
    }
}

// View (EmployeeView class)
class EmployeeView {
    public void printEmployeeDetails(String employeeName, int employeeId, double employeeSalary) {
        System.out.println("Employee Details:");
        System.out.println("Name: " + employeeName);
        System.out.println("Employee ID: " + employeeId);
        System.out.println("Salary: $" + employeeSalary);
    }
}

// Controller (EmployeeController class)
class EmployeeController {
    private Employee model;
    private EmployeeView view;

    public EmployeeController(Employee model, EmployeeView view) {
        this.model = model;
        this.view = view;
    }

    public void setEmployeeName(String name) {
        model.setName(name);
    }

    public String getEmployeeName() {
        return model.getName();
    }

    public void setEmployeeId(int id) {
        model.setEmployeeId(id);
    }

    public int getEmployeeId() {
        return model.getEmployeeId();
    }

    public void setEmployeeSalary(double salary) {
        model.setSalary(salary);
    }

    public double getEmployeeSalary() {
        return model.getSalary();
    }

    public void updateView() {
        view.printEmployeeDetails(model.getName(), model.getEmployeeId(), model.getSalary());
    }
}

public class MVCEmployeeExample {
    public static void main(String[] args) {
        // Create an Employee model
        Employee employee = new Employee();
        employee.setName("John Doe");
        employee.setEmployeeId(101);
        employee.setSalary(60000.0);

        // Create an EmployeeView
        EmployeeView employeeView = new EmployeeView();

        // Create an EmployeeController with the model and view
        EmployeeController employeeController = new EmployeeController(employee, employeeView);

        // Update the model data through the controller
        employeeController.setEmployeeName("Alice Smith");
        employeeController.setEmployeeId(102);
        employeeController.setEmployeeSalary(65000.0);

        // Display the updated data through the view
        employeeController.updateView();
    }
}
```

```
Employee Details:  
Name: Alice Smith  
Employee ID: 102  
Salary: $65000.0  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```