Week 9 – Tutorial Assignment

GRAPHICAL USER INTERFACE BASED PROGRAMMING PARADIGM

Name: M.Jaswanth
Reg No: RA2211003011414

1. Write a java program using swing by inheritance.

Code

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

// CustomFrame class inherits from JFrame
class CustomFrame extends JFrame {
   JButton button;

   public CustomFrame() {
      // Set frame properties
      setTitle("Swing Inheritance Example");
      setSize(300, 200);
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

      // Create a button
      button = new JButton("Click Me");

      // Add action listener to the button
      button.addActionListener(new ActionListener() {
         public void actionPerformed(ActionEvent e) {
            JOptionPane.showMessageDialog(null, "Button Clicked!");
         }
      });

      // Add the button to the frame
      add(button);

      // Set layout (optional)
      setLayout(null);
   }
}

public class Main {
   public static void main(String[] args) {
      // Create an instance of CustomFrame
      CustomFrame frame = new CustomFrame();

      // Set frame visibility
      frame.setVisible(true);
   }
}
```

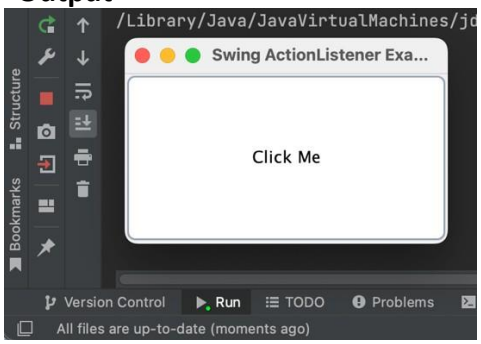
Output

2. Write a java program using swing with ActionListener.

Code

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

public class SwingActionListenerExample
   { public static void main(String[] args) {
      JFrame frame = new JFrame("Swing ActionListener
      Example"); JButton button = new JButton("Click Me");

      // Add an ActionListener to the button
      button.addActionListener(new ActionListener() {
         public void actionPerformed(ActionEvent e)
            { JButton source = (JButton) e.getSource();
            source.setText("Clicked!");
         }
      });

      frame.getContentPane().add(button)
      ; frame.setSize(300, 200);
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      frame.setVisible(true);
   }
}
```

Output



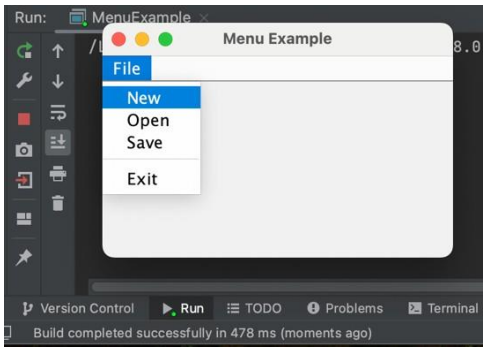
3. Using Java JMenuItem and JMenu implement application swing.

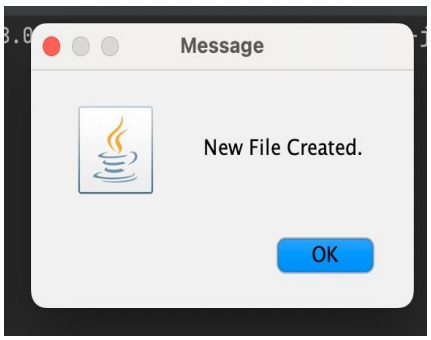
Code

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

public class MenuExample {
   public static void main(String[] args) {
      JFrame frame = new JFrame("Menu Example");

      JMenuBar menuBar = new JMenuBar();
      JMenu fileMenu = new JMenu("File");
      JMenuItem newItem = new JMenuItem("New");
      JMenuItem openItem = new JMenuItem("Open");
      JMenuItem saveItem = new JMenuItem("Save");
      JMenuItem exitItem = new JMenuItem("Exit");

      // Add ActionListeners to menu items
      newItem.addActionListener(new ActionListener() {
         public void actionPerformed(ActionEvent e) {
            // Add your action here
            JOptionPane.showMessageDialog(null, "New File
            Created.");
         }
      });

      openItem.addActionListener(new ActionListener() {
         public void actionPerformed(ActionEvent e) {
            // Add your action here
            JOptionPane.showMessageDialog(null, "Opened File.");
         }
      });

      saveItem.addActionListener(new ActionListener()
         { public void actionPerformed(ActionEvent e) {
            // Add your action here
            JOptionPane.showMessageDialog(null, "File
            Saved.");
         }
      });

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

      fileMenu.add(newItem);
      fileMenu.add(openItem);
```

```
        fileMenu.add(saveItem);
        fileMenu.addSeparator()
        ; fileMenu.add(exitItem);

        menuBar.add(fileMenu);

        frame.setJMenuBar(menuBar);
        frame.setSize(300, 200);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}
```

Output





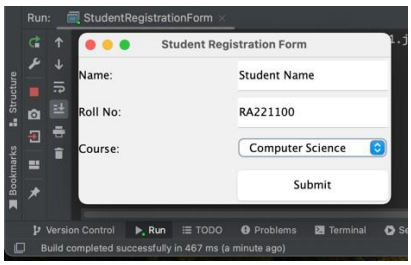
4. Develop a student registration form using SWING components.

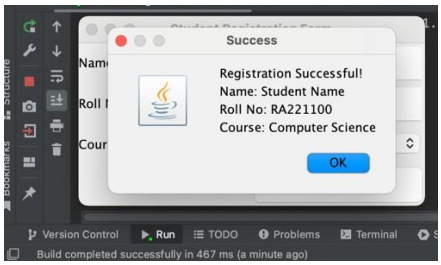
Code

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

public class StudentRegistrationForm {
    public static void main(String[] args) {
        // Create the main frame
        JFrame frame = new JFrame("Student Registration Form");

        // Create labels, text fields, and a button
        JLabel nameLabel = new JLabel("Name:");
        JTextField nameField = new
        JTextField(20);

        JLabel rollLabel = new JLabel("Roll No:");
        JTextField rollField = new JTextField(10);

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

        JButton submitButton = new JButton("Submit");
```

```
        // Add action listener to the button
        submitButton.addActionListener(new ActionListener()
        {
           public void actionPerformed(ActionEvent e) {
              // Retrieve user inputs
              String name =
              nameField.getText(); String rollNo
              = rollField.getText();
              String course = (String) courseComboBox.getSelectedItem();

              // Validate input
              if (name.isEmpty() || rollNo.isEmpty() || course.equals("Select")) {
                 JOptionPane.showMessageDialog(frame, "Please fill out all fields.", "Error",
JOptionPane.ERROR_MESSAGE);
              } else {
                 // Process the registration (in this example, just display a message)
                 String message = String.format("Registration Successful!\nName: %s\nRoll No:
%s\nCourse: %s", name, rollNo, course);
                 JOptionPane.showMessageDialog(frame,          message,          "Success",
JOptionPane.INFORMATION_MESSAGE);
              }
           }
        });

        // Create a panel and add components to it
        JPanel panel = new JPanel(new GridLayout(4,
        2)); panel.add(nameLabel);
        panel.add(nameField);
        panel.add(rollLabel);
        panel.add(rollField);
        panel.add(courseLabel);
        panel.add(courseComboBox);
        panel.add(new JLabel()); // Empty label for spacing
        panel.add(submitButton);

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

        // Set frame properties
        frame.setSize(300, 200);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
     }
}
```

Output





5. Implement Employment registration form using SWING components.

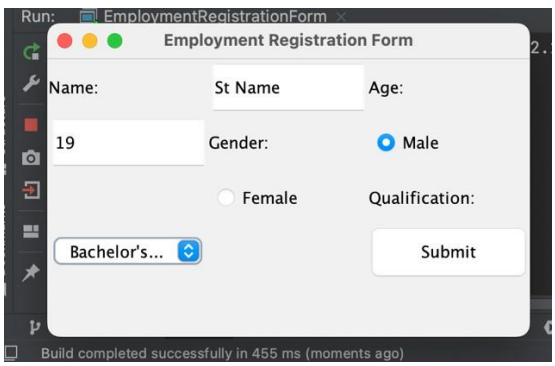
Code

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

public class EmploymentRegistrationForm {
   public static void main(String[] args) {
      // Create the main frame
      JFrame frame = new JFrame("Employment Registration Form");

      // Create labels, text fields, and a button
      JLabel nameLabel = new JLabel("Name:");
      JTextField nameField = new
      JTextField(20);

      JLabel ageLabel = new JLabel("Age:");
      JTextField ageField = new
      JTextField(10);

      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 qualificationLabel = new JLabel("Qualification:");
      String[] qualifications = {"Select", "High School", "Bachelor's Degree", "Master's Degree",
"PhD"};
      JComboBox<String> qualificationComboBox = new JComboBox<>(qualifications);

      JButton submitButton = new JButton("Submit");

      // Add action listener to the button
      submitButton.addActionListener(new ActionListener()
      {
         public void actionPerformed(ActionEvent e) {
            // Retrieve user inputs
            String name =
            nameField.getText(); String age =
            ageField.getText();
            String gender = maleRadioButton.isSelected() ? "Male" : "Female";
```

```
String qualification = (String)
qualificationComboBox.getSelectedItem();
```

```
            // Validate input
            if (name.isEmpty() || age.isEmpty() || qualification.equals("Select") ||
(!maleRadioButton.isSelected() && !femaleRadioButton.isSelected())) {
                JOptionPane.showMessageDialog(frame, "Please fill out all fields.", "Error",
JOptionPane.ERROR_MESSAGE);
            } else {
                // Process the registration (in this example, just display a message)
                String message = String.format("Registration Successful!\nName: %s\nAge:
%s\nGender: %s\nQualification: %s", name, age, gender, qualification);
                JOptionPane.showMessageDialog(frame, message, "Success",
JOptionPane.INFORMATION_MESSAGE);
            }
          }
      });

      // Create a panel and add components to it
      JPanel panel = new JPanel(new GridLayout(5,
      2)); panel.add(nameLabel);
      panel.add(nameField);
      panel.add(ageLabel);
      panel.add(ageField);
      panel.add(genderLabel);
      panel.add(maleRadioButton);
      panel.add(new JLabel()); // Empty label for
      spacing panel.add(femaleRadioButton);
      panel.add(qualificationLabel);
      panel.add(qualificationComboBox);
      panel.add(new JLabel()); // Empty label for
      spacing panel.add(submitButton);

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

      // Set frame properties
      frame.setSize(300, 250);
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      frame.setVisible(true);
    }
}
```

Output

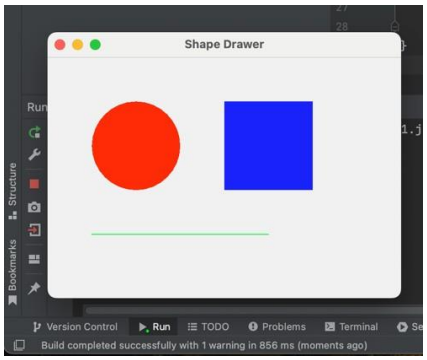


6. Write a java program to draw Oval, Rectangle,Line and fill the color in it.and display it on Applet.

Code

```
import
javax.swing.*;
import java.awt.*;

public class ShapeDrawer extends JApplet {
   public void paint(Graphics g) {
      // Set colors
      g.setColor(Color.red); // Color for oval
      g.fillOval(50, 50, 100, 100); // Draw an oval

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

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

   public static void main(String[] args) {
      JFrame frame = new JFrame("Shape
      Drawer"); ShapeDrawer applet = new
      ShapeDrawer();

      frame.getContentPane().add(applet);
      frame.setSize(400, 300);
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      frame.setVisible(true);

      applet.init();
      applet.start();
   }
}
```

Output



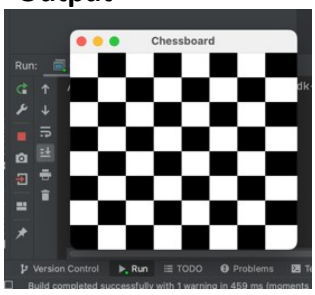
7. Draw a chessboard in java applet.

Code

```
import
javax.swing.*;
import java.awt.*;

public class Chessboard extends JApplet {
   public void paint(Graphics g) {
      int x, y;
      int width = getSize().width / 8;
      int height = getSize().height /
      8;

      for (int row = 0; row < 8; row++)
         { for (int col = 0; col < 8; col++)
         {
            x = col * width; y
            = row * height;

            if ((row % 2 == 0 && col % 2 != 0) || (row % 2 != 0 && col % 2 == 0))
               { g.setColor(Color.black);
            } else {
               g.setColor(Color.white);
            }

            g.fillRect(x, y, width, height);
         }
      }
   }

   public static void main(String[] args) {
      JFrame frame = new
      JFrame("Chessboard"); Chessboard applet
      = new Chessboard();

      frame.getContentPane().add(applet);
      frame.setSize(400, 400);
```

```
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);

        applet.init();
        applet.start();
    }
}
```

Output



8. Write a java program that handles all mouse events and shows the event name at the center of the window when mouse event is fired (Use Adapter classes and applet).

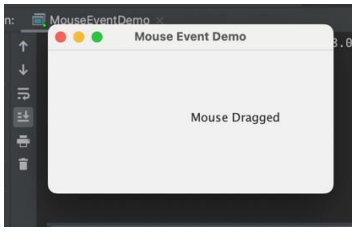
Code

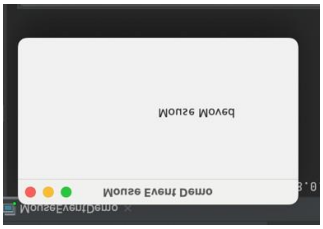
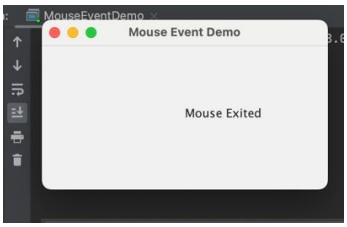
```
import java.awt.*;
import java.awt.event.*;

public class MouseEventDemo extends java.applet.Applet {
    String msg = "";

    public void init() {
        addMouseListener(new MyMouseAdapter(this));
        addMouseMotionListener(new MyMouseMotionAdapter(this));
    }

    public void paint(Graphics g) {
        g.drawString(msg, getWidth()/2, getHeight()/2);
    }

    public static void main(String[] args) {
        Frame frame = new Frame("Mouse Event Demo");
        MouseEventDemo applet = new MouseEventDemo();
        frame.add(applet);
        frame.setSize(400, 300);

        frame.addWindowListener(new WindowAdapter()
            { public void windowClosing(WindowEvent we) {
                frame.dispose();
            }
        });
```

```
        applet.init();
        applet.start();

        frame.setVisible(true);
    }
}

class MyMouseAdapter extends MouseAdapter {
    MouseEventDemo mouseEventDemo;

    public MyMouseAdapter(MouseEventDemo mouseEventDemo)
        { this.mouseEventDemo = mouseEventDemo;
    }

    public void mouseClicked(MouseEvent me) {
        mouseEventDemo.msg = "Mouse Clicked";
        mouseEventDemo.repaint();
    }

    public void mouseEntered(MouseEvent me) {
        mouseEventDemo.msg = "Mouse Entered";
        mouseEventDemo.repaint();
    }

    public void mouseExited(MouseEvent me) {
        mouseEventDemo.msg = "Mouse Exited";
        mouseEventDemo.repaint();
    }

    public void mousePressed(MouseEvent me) {
        mouseEventDemo.msg = "Mouse Pressed";
        mouseEventDemo.repaint();
    }

    public void mouseReleased(MouseEvent me) {
        mouseEventDemo.msg = "Mouse Released";
        mouseEventDemo.repaint();
    }
}

class MyMouseMotionAdapter extends MouseMotionAdapter {
    MouseEventDemo mouseEventDemo;

    public MyMouseMotionAdapter(MouseEventDemo mouseEventDemo)
        { this.mouseEventDemo = mouseEventDemo;
    }
```

```
    public void mouseDragged(MouseEvent me) {
        mouseEventDemo.msg = "Mouse Dragged";
        mouseEventDemo.repaint();
    }

    public void mouseMoved(MouseEvent me) {
        mouseEventDemo.msg = "Mouse Moved";
        mouseEventDemo.repaint();
    }
}
```

Output



9. Implement java MVC pattern application with Student object Model, StudentView and StudentController.

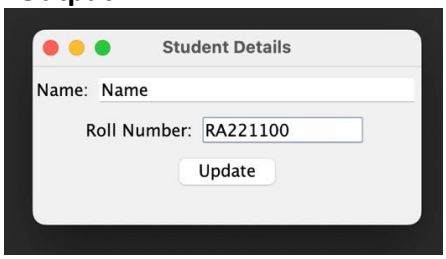
Code

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

public class Main {
    public static void main(String[] args) {
        // Model
        class Student {
            private String name;
            private int rollNumber;

            public String getName() {
                return name;
            }

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

            public int getRollNumber() {
                return rollNumber;
            }

            public void setRollNumber(int rollNumber) {
```

```
        this.rollNumber = rollNumber;
    }
}

// View
class StudentView {
    private JFrame frame;
    private JTextField nameField;
    private JTextField
    rollNumberField; private JButton
    updateButton;

    public StudentView() {
        frame = new JFrame("Student Details");
        nameField = new JTextField(20);
        rollNumberField = new JTextField(10);
        updateButton = new
        JButton("Update");

        frame.setSize(300, 150);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JPanel panel = new JPanel();
        panel.add(new JLabel("Name:"));
        panel.add(nameField);
        panel.add(new JLabel("Roll Number:"));
        panel.add(rollNumberField);
        panel.add(updateButton);

        frame.add(panel);
    }

    public String getNameInput() {
        return nameField.getText();
    }

    public int getRollNumberInput() {
        return Integer.parseInt(rollNumberField.getText());
    }

    public void setUpdateButtonListener(ActionListener listener)
        { updateButton.addActionListener(listener);
    }

    public void displayMessage(String message) {
        JOptionPane.showMessageDialog(frame, message);
    }

    public void clearInputs() {
```

```
            nameField.setText("");
            rollNumberField.setText("")
            ;
        }

        public void show() {
            frame.setVisible(true);
        }
    }

    // Controller
    class StudentController {
        private Student model;
        private StudentView view;

        public StudentController(Student model, StudentView view)
            { this.model = model;
            this.view = view;

            view.setUpdateButtonListener(new UpdateButtonListener());
        }

        class UpdateButtonListener implements ActionListener
            { public void actionPerformed(ActionEvent e) {
                String name = view.getNameInput();
                int rollNumber = view.getRollNumberInput();

                model.setName(name);
                model.setRollNumber(rollNumber);

                view.displayMessage("Student details
                updated!"); view.clearInputs();
            }
        }
    }

    // Create a model, view, and controller
    Student model = new Student();
    StudentView view = new StudentView();
    StudentController controller = new StudentController(model, view);

    // Show the GUI
    view.show();
  }
}
```

Output



10. Implement java MVC to display Employee details.

Code

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

public class Main {
   public static void main(String[] args) {
      // Model
      class Employee {
         private String name;
         private int employeeId;

         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;
         }
      }

      // View
      class EmployeeView {
         private JFrame frame;
         private JTextField nameField;
         private JTextField idField;
         private JButton
         displayButton;

         public EmployeeView() {
            frame = new JFrame("Employee Details");
            nameField = new JTextField(20);
```

```
        idField = new JTextField(10);
        displayButton = new
        JButton("Display");

        frame.setSize(300, 150);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JPanel panel = new JPanel();
        panel.add(new JLabel("Name:"));
        panel.add(nameField);
        panel.add(new JLabel("Employee ID:"));
        panel.add(idField);
        panel.add(displayButton);

        frame.add(panel);
    }

    public String getNameInput() {
        return nameField.getText();
    }

    public int getEmployeeIdInput() {
        return Integer.parseInt(idField.getText());
    }

    public void setDisplayButtonListener(ActionListener listener)
        { displayButton.addActionListener(listener);
    }

    public void displayEmployeeDetails(String name, int id) {
        JOptionPane.showMessageDialog(frame, "Name: " + name + "\nEmployee ID: " + id);
    }

    public void clearInputs() {
        nameField.setText("");
        idField.setText("");
    }

    public void show() {
        frame.setVisible(true);
    }
}

// Controller
class EmployeeController {
    private Employee model;
    private EmployeeView
    view;
```

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

           view.setDisplayButtonListener(new DisplayButtonListener());
        }

        class DisplayButtonListener implements ActionListener
           { public void actionPerformed(ActionEvent e) {
              String name =
              view.getNameInput(); int id =
              view.getEmployeeIdInput();

              model.setName(name);
              model.setEmployeeId(id);

              view.displayEmployeeDetails(model.getName(),
              model.getEmployeeId()); view.clearInputs();
           }
        }
     }

     // Create a model, view, and controller
     Employee model = new Employee();
     EmployeeView view = new EmployeeView();
     EmployeeController controller = new EmployeeController(model, view);

     // Show the GUI
     view.show();
  }
}
```

Output