Core Java Practical 1

1.A : Write a program to create a class and implement a default, overloaded and copy Constructor.

```
class MyClass
{
private int a;
public MyClass()
{
System.out.println("Default Constructor");
}
public MyClass(int value)
{
a=value;
System.out.println("Parameterized Construsctor and value is: "+a);
}
public MyClass(MyClass other)
{
a = other.a;
System.out.println("Copy Constructor and value is: "+a);
}
}
public class Pr
{
public static void main (String[] args)
{
MyClass obj1 = new MyClass();
MyClass obj2 = new MyClass(7);
MyClass obj3 = new MyClass(obj2);
}
```

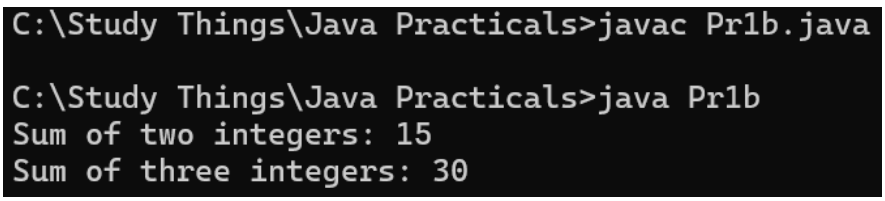
Output:

```
C:\Study Things\Java Practicals>java Pr
Default Constructor
Parameterized Construsctor and value is: 7
Copy Constructor and value is: 7
```

1.B: Write a program to create a class and implement the concepts of method Overloading.

```
class OperOver
{
public int add(int a, int b)
{
return a + b;
}

public int add(int a, int b, int c)
{
return a + b + c;
}
}
public class Pr1b
{
public static void main(String[]args)
{
OperOver obj = new OperOver();

int sum1 = obj.add(5,10);
int sum2 = obj.add(5,10,15);

System.out.println("Sum of two integers: " +sum1);
System.out.println("Sum of three integers: " +sum2);
}
}
```

Output:



1.C: Write a program to create a class and implement the concepts of Static methods.

```
class DemoStaticMethods
{
public static int add(int a, int b)
{
return a + b;
}
public static int subtract(int a, int b)
{
return a - b;
}
}
public class Pr1c
{
public static void main(String[]args)
{
int sum = DemoStaticMethods.add(8,4);
int difference = DemoStaticMethods.subtract(7,6);
System.out.println("Sum: " + sum);
System.out.println("Difference: " + difference);
}
}
```

Output:

```
C:\Study Things\Java Practicals>javac Pr1c.java

C:\Study Things\Java Practicals>java Pr1c
Sum: 12
Difference: 1
```

Core Java Practical 2

2.A: Write a program to implement the concepts of Inheritance and method overriding.

```
class A
{
void show()
{
System.out.println("Base Class");
}
}
class B extends A
{
void show()
{
System.out.println("Derieved Class");
}
}
class Pr2a
{
public static void main(String args[])
{
B s=new B();
s.show();
}
}
```

Output:

```
C:\Study Things\Java Practicals>javac Pr2a.java

C:\Study Things\Java Practicals>java Pr2a
Derieved Class
```

2.B: Write a program to implement the concepts of Abstract classes and methods.

```
abstract class Shape
{
public abstract double area();
}
class Circle extends Shape
{
private double radius;
public Circle(double radius)
{
this.radius = radius;
}
@Override
public double area()
{
return Math.PI*radius*radius;
}
}
public class Pr2b
{
public static void main(String[] args)
{
Circle circle = new Circle(10.0);
System.out.println("Circle Area: "+circle.area());
}
}
```

Output:

```
C:\Study Things\Java Practicals>javac Pr2b.java

C:\Study Things\Java Practicals>java Pr2b
Circle Area: 314.1592653589793
```

2.C: Write a program to implement the concept of interfaces.

```
Interface Shape
{
double area();
double perimeter();
}
class Circle implements Shape
{
private double radius;
public Circle(double radius)
{
this.radius = radius;
}
@Override
public double area()
{
return Math.PI*radius* radius;
}
@Override
public double perimeter()
{
return 2 *Math.PI* radius;
}
}
public class Pr2c
{
public static void main(String[] args)
{
Circle circle = new Circle(10.0);
System.out.println("Circle Area: " + circle.area());
System.out.println("Circle Perimeter:" + circle.perimeter());
}
```

}

Output:

```
C:\Study Things\Java Practicals>javac Pr2c.java

C:\Study Things\Java Practicals>java Pr2c
Circle Area: 314.1592653589793
Circle Perimeter:62.83185307179586
```

Core java Practical 3

3.A: Write a program to raise built-in exceptions and raise them as per the requirements.

```
public class Pr3a
{
public static void main(String[] args)
{
try
{
int result = divide(10,0);
System.out.println("Result: " + result);
}
catch (ArithmeticException e)
{
System.err.println("Error: Division by zero.");
}
}
public static int divide(int a, int b)
{
return a / b;
}
}
```

Output:

```
C:\Study Things\Java Practicals>javac Pr3a.java

C:\Study Things\Java Practicals>java Pr3a
Error: Division by zero.
```

3.B: Write a program to define user defined exceptions and raise them as per the requirements.

```
class CustomException extends Exception
{
public CustomException(String message)
{
super(message);
}
}
public class Pr3b
{
public static void main(String[] args)
{
try
{
int age = -20;
if (age < 0)
{
throw new CustomException("Age cannot be negative.");
}
System.out.println("Age: " + age);
}
catch (CustomException e)
{
System.err.println("Error: " + e.getMessage());
}
}
}
```

Output:

```
C:\Study Things\Java Practicals>javac Pr3b.java

C:\Study Things\Java Practicals>java Pr3b
Error: Age cannot be negative.
```

Core java Practical 4

4. Write the java application to demonstrate multiple bouncing balls of different colours using thread.

```
import java.awt.*;
import java.awt.event.*;
class Bouncing_Balls extends Frame implements MouseListener {
   // initializing co-ordinates
   int x = 40, y = 40, t1 = 1, t2 = 1;
   int x1 = 200, y1 = 40, t12 = 1, t22 = 1;
   int x2 = 100, y2 = 100, t13 = 1, t23 = 1;
   Thread th;

   Bouncing_Balls() {
      setSize(700, 800);
      setVisible(true);
      th = new Thread(new Thread() {
         public void run() {
            while (true) {
               x = x + t1;
               y = y + t2;
               x1 = x1 + t12;
               y1 = y1 + t22;
               x2 = x2 - t13;
               y2 = y2 - t23;
// specifying some condition to make balls move in a particular
               if (x < 0 || x > 680)
                  t1 = t1 * (-1);
               if (y < 20 || y > 780)
                  t2 = t2 * (-1);
               if (x1 < 0 || x1 > 680)
                  t12 = t12 * (-1);
               if (y1 < 20 || y1 > 780)
                  t22 = t22 * (-1);
```

```
            if (x2 < 0 || x2 > 680)
               t13 = t13 * (-1);
            if (y2 < 20 || y2 > 780)
               t23 = t23 * (-1);
            try {
               this.sleep(5);
            } catch (Exception E) {
            }
            repaint();}}}
   );
   addMouseListener(this);}

public void mouseClicked(MouseEvent M) {
   th.start();
}

public void mousePressed(MouseEvent M) {
}

public void mouseReleased(MouseEvent M) {
}

public void mouseEntered(MouseEvent M) {
}

public void mouseExited(MouseEvent M) {
}

public void paint(Graphics g) {
   g.setColor(Color.pink);
   g.fillOval(x, y, 40, 40);
   g.setColor(Color.pink);
```

```
        g.fillOval(x1, y1, 40, 40);
        g.setColor(Color.pink);
        g.fillOval(x2, y2, 40, 40);
    }

    public static void main(String[] args) {
        Bouncing_Balls B = new Bouncing_Balls();
    }
}
```

Output:

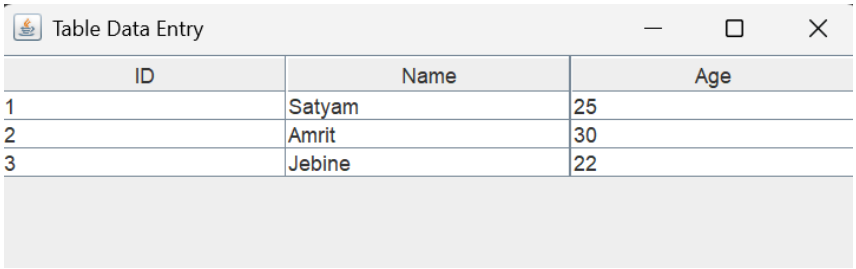
Core java Practical 5

5.A: Write a JDBC program that displays the data of a given table in a GUI Table.

```
import java.sql.*;
import javax.swing.JFrame;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;
public class JDBCGUITableExample {
public static void main(String[] args) {
JFrame frame = new JFrame("Database Table Display");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
String url = "jdbc:mysql://localhost:3306/syit";
String username = "root";
String password = "root";
String tableName = "students";
DefaultTableModel tableModel = new DefaultTableModel();
tableModel.addColumn("ID");
tableModel.addColumn("Name");
tableModel.addColumn("Age");
JTable jTable = new JTable(tableModel);
JScrollPane jScrollPane = new JScrollPane(jTable);
frame.getContentPane().add(jScrollPane);
try {
Connection connection = DriverManager.getConnection(url,
username, password);
Statement statement= connection.createStatement();
String query = "SELECT * FROM " + tableName;
ResultSet resultSet = statement.executeQuery(query);
while (resultSet.next()) {
Object[] row = new Object[3];
row[0] = resultSet.getObject(1);
row[1] = resultSet.getObject(2);
```

```
row[2] = resultSet.getObject(3);
tableModel.addRow(row);
}
resultSet.close();
statement.close();
connection.close();
} catch (Exception e) {
e.printStackTrace();
}
frame.setSize(400, 300);
frame.setLocationRelativeTo(null);
frame.setVisible(true);
}
}
```

Output:



5.B: Write a JDBC program to Show the details of a specified product from a given table selected using Combobox.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;
public class UserDetails extends JFrame{
private JComboBox<String> userComboBox;
private JTextField idTextField;
private JTextField nameTextField;
public UserDetails(){
setTitle("Students");
setSize(400,150);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setLayout(new FlowLayout());
userComboBox= new JComboBox<>();
idTextField = new JTextField(20);
nameTextField= new JTextField(10);
add(new JLabel("Select ID"));
add(userComboBox);
add(new JLabel("Id:"));
add(idTextField);
add(new JLabel("Name"));
add(nameTextField);
try{
Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/syit","root","root");
String query ="SELECT id FROM students";
PreparedStatement preparedStatement= connection.prepareStatement(query);
ResultSet resultSet=preparedStatement.executeQuery();
while (resultSet.next())
{
```

```
int id=resultSet.getInt("id");
userComboBox.addItem(Integer.toString(id));
}
resultSet.close();
preparedStatement.close();
connection.close();
}
catch (SQLException e){
e.printStackTrace();
}
userComboBox.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent e) {
String selectedId = (String) userComboBox.getSelectedItem();
if(selectedId!=null)
{
try{
Connection
connection=DriverManager.getConnection("jdbc:mysql://localhost:3306/syit","root","root");
String query ="SELECT id,name FROM students WHERE id=?";
PreparedStatement preparedStatement=connection.prepareStatement(query);
preparedStatement.setString(1,selectedId);
ResultSet resultSet=preparedStatement.executeQuery();
if (resultSet.next()){
int id = resultSet.getInt("id");
String name = resultSet.getString("name");
idTextField.setText(Integer.toString(id));
nameTextField.setText(name);
}
resultSet.close();
preparedStatement.close();
connection.close();
}catch (SQLException ex){
```

```
ex.printStackTrace();
}
}}}); }
public static void main(String[] args){
SwingUtilities.invokeLater(()->{
UserDetails user = new UserDetails();
user.setVisible(true);
});}}
```

Output:

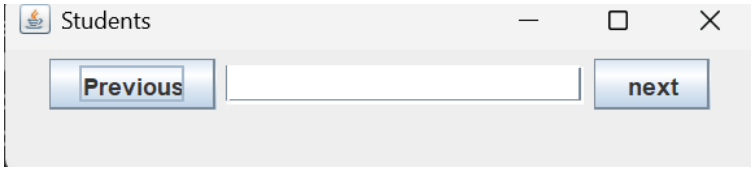
5.C: Write a GUI application to Navigate forward and reverse result set data.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;
public class ForwardBackward extends JFrame{
private JButton previousButton;
private JButton nextButton;
private JTextField dataField;
private ResultSet resultSet;
public ForwardBackward() {
setTitle("Students");
setSize(400, 100);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setLayout(new FlowLayout());
previousButton = new JButton("Previous");
nextButton = new JButton("next");
dataField = new JTextField(20);
add(previousButton);
add(dataField);
add(nextButton);
try {
Connection connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/syit", "root",
"root");
Statement statement = connection.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);
String query = "SELECT * FROM students";
resultSet = statement.executeQuery(query);
displayData();
} catch (SQLException e) {
e.printStackTrace();
}
```

```
previousButton.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent e) {
try {
if (resultSet.previous()){
displayData();
}
}
catch (SQLException ex){
ex.printStackTrace();
}
}
});
nextButton.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent e) {
try {
if (resultSet.next()){
displayData();
}
}
catch (SQLException ex){
ex.printStackTrace();
}
}
});
}
private void displayData(){
try {
dataField.setText(resultSet.getString("name"));
}
catch (SQLException e){
```

```
e.printStackTrace();
}
}
public static void main(String[] args){
SwingUtilities.invokeLater(()->{
ForwardBackward app = new ForwardBackward();
app.setVisible(true);
});
}
}
```

Output:



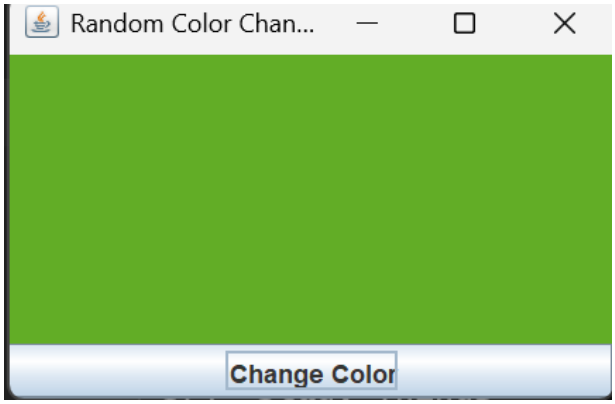
Core java practical 6

6.A: Create a swing application that randomly changes color on button click.

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

public class ChangeColor extends JFrame
{
private JPanel colorPanel;
private JButton changeColorButton;

public ChangeColor()
{
setTitle("Random Color Change");
setSize(300,200);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setLayout(new BorderLayout());

colorPanel = new JPanel();
changeColorButton = new JButton("Change Color");
add(colorPanel, BorderLayout.CENTER);
add(changeColorButton, BorderLayout.SOUTH);

changeColorButton.addActionListener(new ActionListener(){
@Override
public void actionPerformed(ActionEvent e) {
changeColor();
}
});
}
```

```
private void changeColor() {
Random random = new Random();
Color randomColor = new Color(random.nextInt(156), random.nextInt(256), random.nextInt(256));
colorPanel.setBackground(randomColor);
}
public static void main(String[] args) {
SwingUtilities.invokeLater(() -> {
ChangeColor app = new  ChangeColor();
app.setVisible(true);
});
}
}
```

Output:

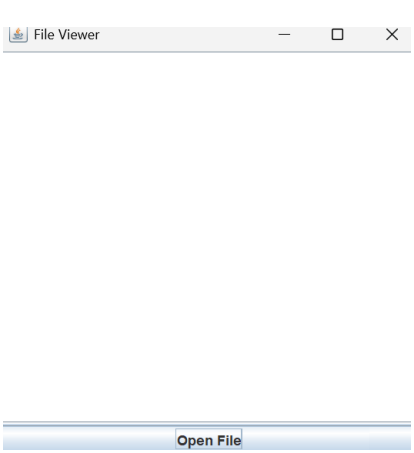


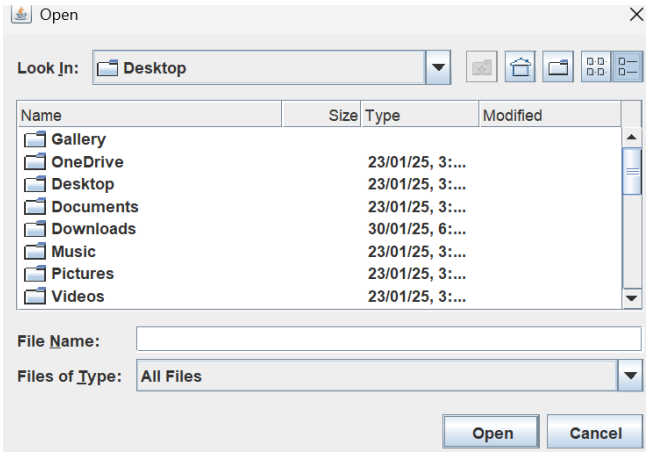
6.B: Create a Swing application to demonstrate use of TextArea using scrollpane to show contest of text file in textarea selected using file chooser.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class ShowFileInTextArea extends JFrame
{
private JTextArea textArea = new JTextArea(20,40);
private JButton openFileButton = new JButton("Open File");

public ShowFileInTextArea()
{
setTitle("File Viewer");
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setLayout(new BorderLayout());
JScrollPane scrollPane = new JScrollPane(textArea);
add(scrollPane, BorderLayout.CENTER);
add(openFileButton, BorderLayout.SOUTH);
openFileButton.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent e) {
openFile();
}
});
pack();
setLocationRelativeTo(null);
}
private void openFile() {
```

```
JFileChooser fileChooser = new JFileChooser();
int result = fileChooser.showOpenDialog(this);
if (result == JFileChooser.APPROVE_OPTION) {
try (BufferedReader reader = new BufferedReader
(new FileReader(fileChooser.getSelectedFile()))) {
StringBuilder Content = new StringBuilder();
String line;
while ((line = reader.readLine()) != null); {
Content.append(line).append("\n");
}
textArea.setLineWrap(true);
textArea.setText(Content.toString());
} catch (IOException e) {
JOptionPane.showMessageDialog(this, "Error reading the file.", "Error",
JOptionPane.ERROR_MESSAGE);
}
}
}
public static void main(String[] args) {
SwingUtilities.invokeLater(() -> {
new ShowFileInTextArea().setVisible(true);
});
}
}
```

Output:



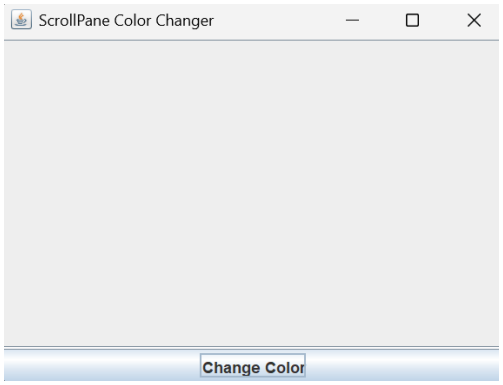


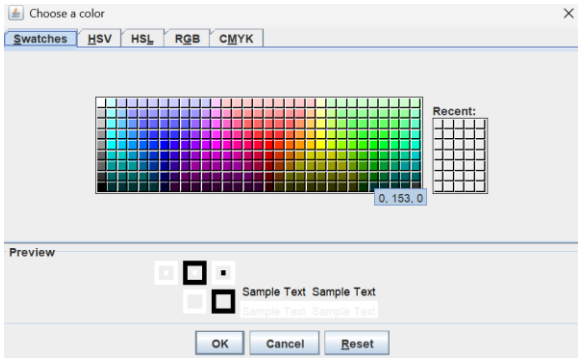
6.C: Create a Swing application to demonstrate use of scrollpane to change its color selected using colour chooser.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
public class ColorChanger extends JFrame
{
private JScrollPane scrollPane = new JScrollPane();
private JButton changeColorButton = new JButton("Change Color");
public ColorChanger() {
setTitle("ScrollPane Color Changer");
setSize(400,300);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setLayout(new BorderLayout());
add(scrollPane, BorderLayout.CENTER);
add(changeColorButton, BorderLayout.SOUTH);
changeColorButton.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent e) {
Color selectedColor = JColorChooser.showDialog
(null, "Choose a color",scrollPane.getBackground());
if (selectedColor != null) {
scrollPane.getViewport().setBackground(selectedColor);
}
}
});
}
public static void main(String[] args) {
SwingUtilities.invokeLater(() -> {
new ColorChanger().setVisible(true);
});
}
```

}

Output:


ScrollPane Color Changer
Change Color

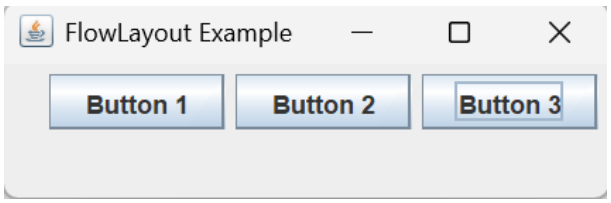

Choose a color
Swatches
HSV
HSL
RGB
CMYK
Recent:
0, 153, 0
Preview
Sample Text Sample Text
OK
Cancel
Reset

Core java practical 7

7.A: Write a program on Flow Layout.

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

public class DemoFlowLayout
{
public static void main(String[] args)
{
JFrame frame = new JFrame("FlowLayout Example");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setSize(300,100);

JPanel panel = new JPanel(new FlowLayout(FlowLayout.RIGHT));

JButton button1 = new JButton("Button 1");
JButton button2 = new JButton("Button 2");
JButton button3 = new JButton("Button 3");

panel.add(button1);
panel.add(button2);
panel.add(button3);

frame.add(panel);
frame.setVisible(true);
}
}
```

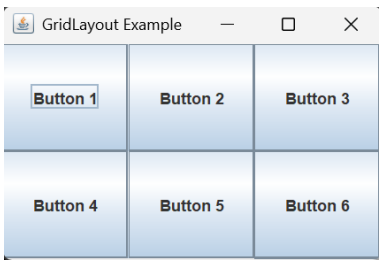
Output:



7.B:Write a program on Gride Layout.

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

public class DemoGridLayout {
public static void main(String[] args) {
JFrame frame = new JFrame("GridLayout Example");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setSize(300,200);
JPanel panel = new JPanel(new GridLayout(2,3));

JButton button1 = new JButton("Button 1");
JButton button2 = new JButton("Button 2");
JButton button3 = new JButton("Button 3");
JButton button4 = new JButton("Button 4");
JButton button5 = new JButton("Button 5");
JButton button6 = new JButton("Button 6");

panel.add(button1);
panel.add(button2);
panel.add(button3);
panel.add(button4);
panel.add(button5);
panel.add(button6);

frame.add(panel);
frame.setVisible(true);
} }
```

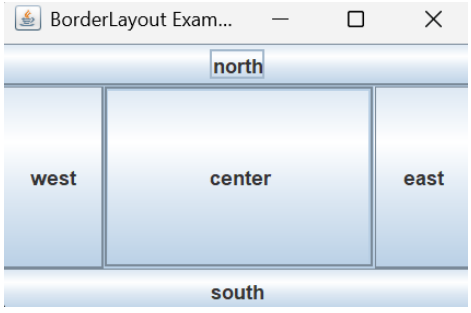
Output:



7.C: Write a program on Border Layout.

```
import javax.swing.*;
import java.awt.*;
public class DemoBorderLayout
{
public static void main(String[] args)
{
JFrame frame = new JFrame("BorderLayout Example");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setSize(300,200);

JButton northButton = new JButton("north");
JButton southButton = new JButton("south");
JButton eastButton = new JButton("east");
JButton westButton = new JButton("west");
JButton centerButton = new JButton("center");

Container contentPane = frame.getContentPane();
contentPane.setLayout(new BorderLayout());

contentPane.add(northButton, BorderLayout.NORTH);
contentPane.add(southButton, BorderLayout.SOUTH);
contentPane.add(eastButton, BorderLayout.EAST);
contentPane.add(westButton, BorderLayout.WEST);
contentPane.add(centerButton, BorderLayout.CENTER);
frame.setVisible(true);
}
```

Output:

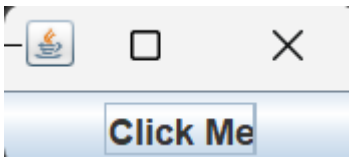


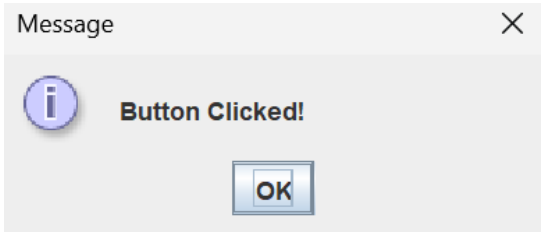
Core java practical 8

8.A.1: Write a program to demonstrate the Action Event.

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

public class BtnClkDemo {
public static void main(String[] args) {
JFrame frame = new JFrame("Button Click Demo");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

JButton button = new JButton("Click Me");
button.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent e) {
JOptionPane.showMessageDialog(frame,"Button Clicked!");
}
});
frame.getContentPane().add(button);
frame.pack();
frame.setVisible(true);
}
}
```

Output:

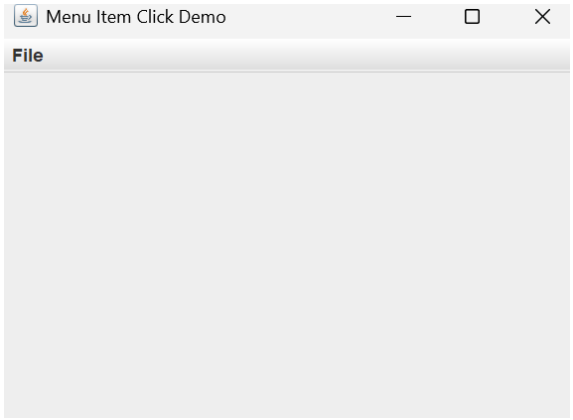


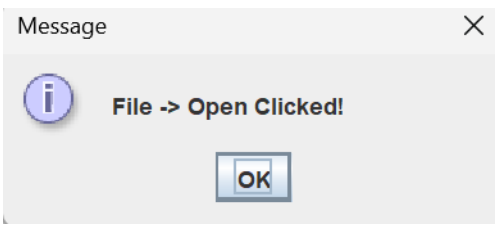


8.A.2: Write a program to demonstrate the Action Event with Menu Item.

```
import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
public class MenuItmClk {
public static void main(String[] args) {
JFrame frame = new JFrame("Menu Item Click Demo");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
JMenuBar menuBar = new JMenuBar();
JMenu fileMenu = new JMenu("File");
JMenuItem openItem = new JMenuItem("Open");
openItem.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent e) {
JOptionPane.showMessageDialog(frame,"File -> Open Clicked!");
}
});
fileMenu.add(openItem);
menuBar.add(fileMenu);
frame.setJMenuBar(menuBar);
frame.setSize(400,300);
frame.setVisible(true);
}
}
```

Output:

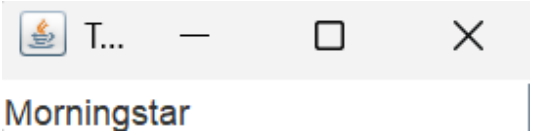


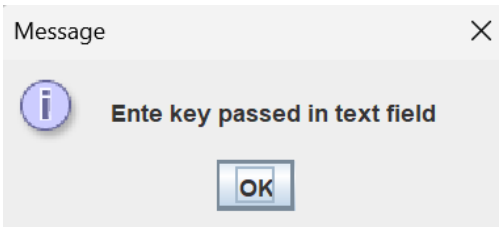


8.A.3: Write a program to demonstrate the Action Event with Text Field Enter Key Event.

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

public class TxtEntrKey {
public static void main(String[] args) {
JFrame frame = new JFrame("Text Field Enter Key Demo");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

JTextField textField = new JTextField(20);

textField.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent e) {
JOptionPane.showMessageDialog(frame,"Enter key passed in text field");
}
});
frame.getContentPane().add(textField);
frame.pack();
frame.setVisible(true);
}
}
```

Output:



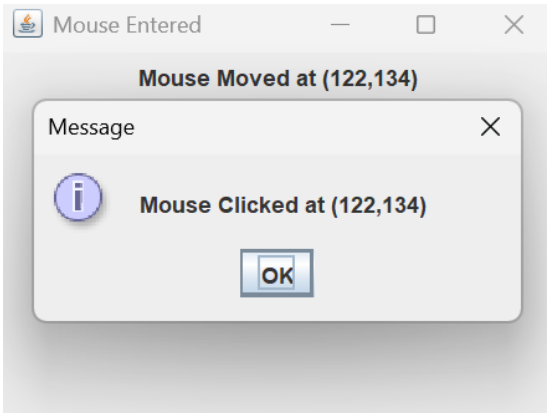


8.B: Program to demonstrate MouseClick, MouseMotion and MouseEnter/Exit events.

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

public class AllMouseEvents
{
public static void main(String[] args)
{
SwingUtilities.invokeLater(() -> {
JFrame frame = new JFrame("Mouse Click Event Demo");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

JPanel panel = new JPanel();
JLabel jl = new JLabel();
panel.add(jl);
panel.setPreferredSize(new Dimension(300,200));

panel.addMouseListener(new MouseAdapter() {
@Override
public void mouseClicked(MouseEvent e) {
JOptionPane.showMessageDialog(frame,"Mouse Clicked at (" + e.getX() + "," + e.getY() + ")");
}
});

panel.addMouseMotionListener(new MouseAdapter() {
@Override
public void mouseMoved(MouseEvent e) {
jl.setText("Mouse Moved at (" + e.getX() + "," + e.getY() + ")");
```

```
}
});

panel.addMouseListener(new MouseAdapter() {
@Override
public void mouseEntered(MouseEvent e) {
frame.setTitle("Mouse Entered");
}
@Override
public void mouseExited(MouseEvent e) {
frame.setTitle("Mouse Exited");
}
});
frame.add(panel);
frame.pack();
frame.setVisible(true);
});
}
}
```

Output:



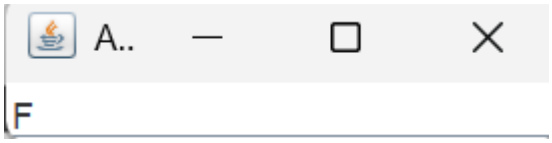
8.C: Program to demonstrates how to handle KeyEvents.

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

public class AllKeyEvents
{
public static void main(String[] args)
{
JFrame frame = new JFrame("All KeyEvents");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

JTextField textField = new JTextField(20);
frame.add(textField);

textField.addKeyListener(new KeyAdapter() {
@Override
public void keyTyped(KeyEvent e) {
System.out.println("Key Typed: " + e.getKeyChar());
}

public void KeyPressed(KeyEvent e)
{
System.out.println("Key Pressed: " + KeyEvent.getKeyText(e.getKeyCode()));
}
@Override
public void keyReleased(KeyEvent e)
{
System.out.println("Key Released: " + KeyEvent.getKeyText(e.getKeyCode()));
}
});

frame.pack();
```

```
frame.setVisible(true);

}

}
```

Output:

```
C:\Study Things\Java Practicals>java AllKeyEvents
Key Typed: F
Key Released: F
```

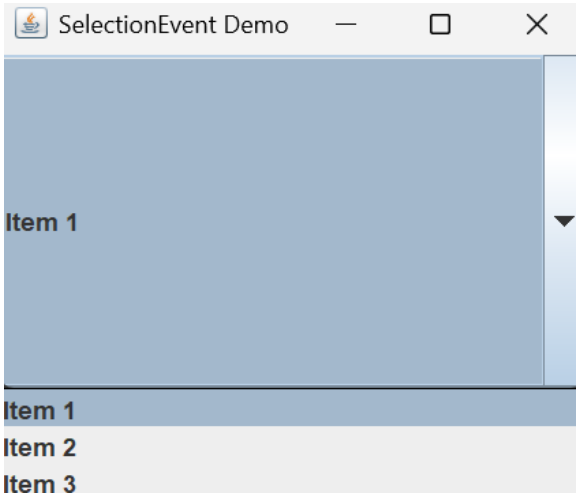


Code 8.D: Program to demonstrate Selection Event.

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

public class SelectionEvents
{
public static void main(String[] args)
{
SwingUtilities.invokeLater(() -> {
JFrame frame = new JFrame("SelectionEvent Demo");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

String[] items = {"Item 1", "Item 2", "Item 3"};

JComboBox<String> comboBox = new JComboBox<>(items);
comboBox.addActionListener(new ActionListener()
{
@Override
public void actionPerformed(ActionEvent e)
{
String selectedItem = (String) comboBox.getSelectedItem();
System.out.println("Selected: " + selectedItem);
}
});

frame.add(comboBox, BorderLayout.CENTER);
frame.setSize(300,200);
frame.setVisible(true);
});
}
```

}

Output:

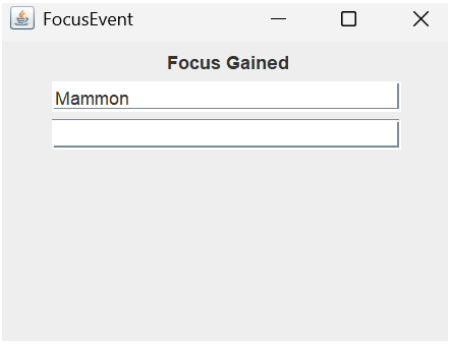


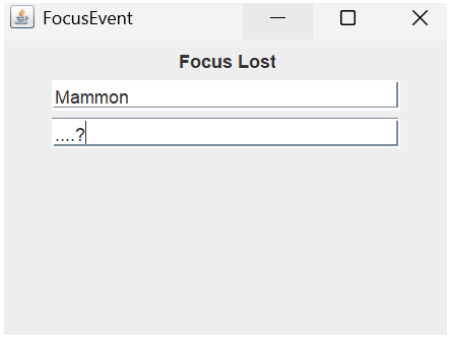
8.E: Program to demonstrates Focus Lost and Focus Gained.

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

public class AllFocusEvents
{
public static void main(String[] args)
{
SwingUtilities.invokeLater(() -> {
JFrame frame = new JFrame("FocusEvent");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

JPanel panel = new JPanel();
panel.setPreferredSize(new Dimension(300,200));

JLabel label = new JLabel();
JTextField textField1 = new JTextField(25);
JTextField textField2 = new JTextField(25);

textField1.addFocusListener(new FocusListener()
{
@Override
public void focusGained(FocusEvent e)
{
label.setText("Focus Gained");
}
@Override
public void focusLost(FocusEvent e)
{
label.setText("Focus Lost");
```

```
}
});
panel.add(label);
panel.add(textField1);
panel.add(textField2);
frame.add(panel);
frame.pack();
frame.setVisible(true);
});
}
}
```

Output:



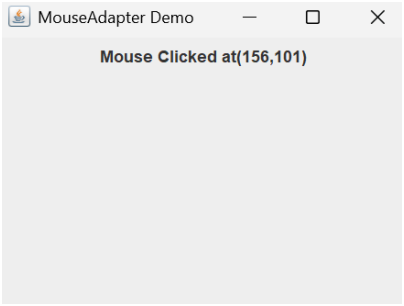


Core java practical 9

9: Demonstrate the use of Adapter Class in Event Handling.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.MouseEvent;
import java.awt.event.MouseAdapter;
public class MyAdapterClass
{
public static void main(String[] args)
{
SwingUtilities.invokeLater(() -> {
JFrame frame = new JFrame("MouseAdapter Demo");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
JLabel label = new JLabel();
JPanel panel = new JPanel();
panel.setPreferredSize(new Dimension(300,200));
panel.addMouseListener(new MouseAdapter() {
@Override
public void mouseClicked(MouseEvent e) {
label.setText("Mouse Clicked at(" + e.getX() + "," +e.getY() + ")");
}
});
panel.add(label);
frame.add(panel);
frame.pack();
frame.setVisible(true);
});
}}
}
```

Output:



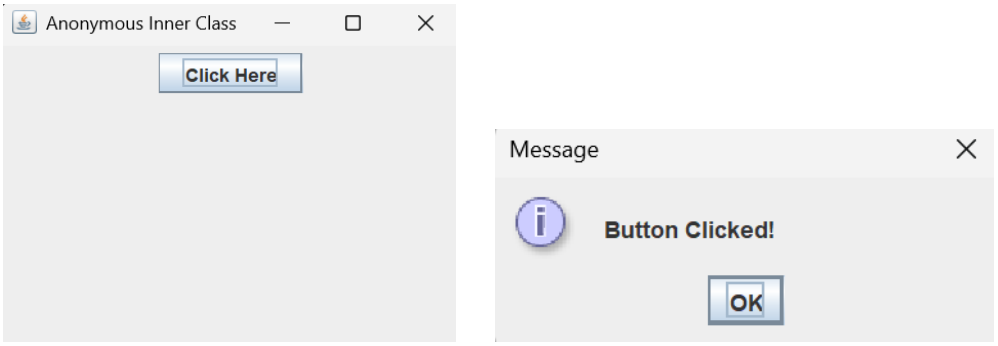
Core java practical 10

10: Demonstrate the use of Aninymous Inner Class in Event Handling.

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

public class MyAnonymousInner
{
public static void main(String[] args)
{
SwingUtilities.invokeLater(() -> {
JFrame frame = new JFrame("Anonymous Inner Class");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

JPanel panel = new JPanel();
panel.setPreferredSize(new Dimension(300,200));

JButton button = new JButton("Click Here");

button.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent e) {
JOptionPane.showMessageDialog(frame, "Button Clicked!");
}
});
panel.add(button);
frame.add(panel);
frame.pack();
frame.setVisible(true);
});
}
```

}

Output:


Anonymous Inner Class
Click Here
Message
Button Clicked!
OK