**SATHYABMA INSTITUTE OF SCIENCE AND TECHNOLOGY SCHOOL OF COMPUTING**

**S12BLH31 PROGRAMMING IN JAVA**

**Unit – V**

#### 5 a. Creating Frame Window by Instantiating Frame class

**Algorithm:**

Step 1: **Start.**

Step 2: **Create a class named Testawt.**

Step 3: **Define the constructor Testawt()**:

* **Step 3.1**: Inside the constructor, create a Frame object named fm.
* **Step 3.2**: Create a Label object named lb with the text "welcome to java graphics".
* **Step 3.3**: Add the Label to the Frame using the add() method.
* **Step 3.4**: Set the size of the Frame to 300x300 using the setSize() method.
* **Step 3.5**: Make the Frame visible using the setVisible() method with true as the argument.

Step 4: **In the main() method**:

* **Step 4.1**: Create an object of the Testawt class, which will automatically call the constructor and display the frame.

Step 5: **End.**

**Program:**

import java.awt.\*;

public class Testawt

{

Testawt()

Frame fm=new Frame(); //Creating a frame

Label lb = new Label("welcome to java graphics");

fm.add(lb); //adding label to the frame

fm.setSize(300, 300); //setting frame size

fm.setVisible(true) ;

}

public static void main(String args[])

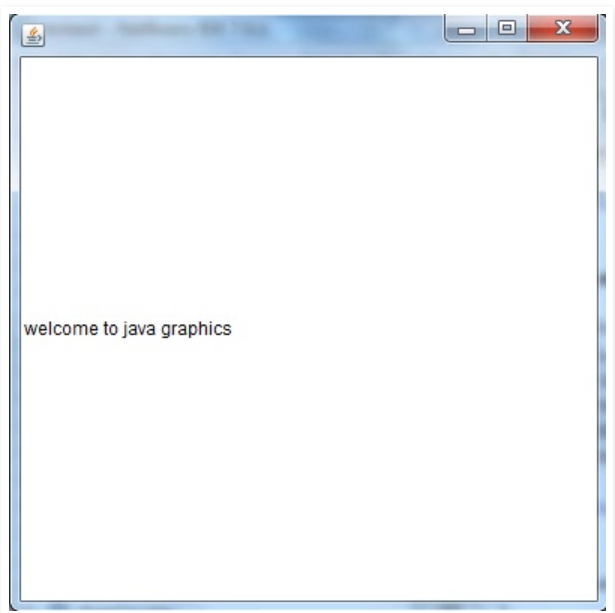
{

Testawt ta = new Testawt();

}

}

Sample Output:



5 b. Write a Java program to generate simple calculator application using AWT.

**Algorithm for AWT Calculator**

**Step 1: Start**

**Step 2: Initialize Components**:

* + Create a **Frame** f1.
  + Create three **Labels**: l1, l2, l3 for "First Number", "Second Number", and "Result".
  + Create three **TextFields**: t1, t2 for user input, and t3 for displaying the result.
  + Create five **Buttons**: b1 for "Add", b2 for "Subtract", b3 for "Multiply", b4 for "Divide", and b5 for "Cancel".

**Step 3: Set Layout**:

* + Set the layout of the frame as **GridLayout** to arrange components in a structured way (6 rows x 2 columns).

**Step 4: Add Components**:

* + Add all components (Labels, TextFields, and Buttons) to the frame in the following order:
    - Label l1, TextField t1 (for first number).
    - Label l2, TextField t2 (for second number).
    - Label l3, TextField t3 (for displaying the result, set t3 as non-editable).
    - Buttons b1, b2, b3, b4, b5.

**Step 5: Implement Button Operations**:

* + **For Add Button (b1)**:
    - Add an **ActionListener** to capture button click.
    - Retrieve input from t1 and t2, convert to integers.
    - Add the two numbers and display the result in t3.
  + **For Subtract Button (b2)**:
    - Add an **ActionListener**.
    - Retrieve input from t1 and t2, convert to integers.
    - Subtract the second number from the first and display the result in t3.
  + **For Multiply Button (b3)**:
    - Add an **ActionListener**.
    - Retrieve input from t1 and t2, convert to integers.
    - Multiply the two numbers and display the result in t3.
  + **For Divide Button (b4)**:
    - Add an **ActionListener**.
    - Retrieve input from t1 and t2, convert to integers.
    - Check if the second number (t2) is zero:
      * If it is zero, display "Error: Div by 0" in t3.
      * Otherwise, divide the first number by the second and display the result in t3.
  + **For Cancel Button (b5)**:
    - Add an **ActionListener**.
    - On button click, close the application (dispose the frame).

**Step 6: Display Frame**:

* + Set the frame to be visible so that users can interact with the GUI.

**Step 7: Handle Window Close**:

* + Add a **WindowListener** to close the frame when the user clicks the window close button.

**Step 8: End**

**Program:**

// simple calculator application using AWT

import java . awt . \* ;

import java . awt . event . \* ;

class simplecalculator implements ActionListener {

Frame f1 = new Frame ( ) ;

Label l1= new Label ( "First Number") ;

Label l2=new Label ( "Second Number") ;

Label l3= new Label ("Result") ;

TextField t1 = new TextField() ;

TextField t2 = new TextField() ;

TextField t3 = new TextField() ;

Button b1= new Button("Add+");

Button b2= new Button("Sub-");

Button b3= new Button("Mul\*");

Button b4= new Button("Div/");

Button b5= new Button("Cancel");

simplecalculator( ) {

//Giving Coordinates

11.setBounds (40, 100, 100, 15);

12.setBounds (40, 140, 100, 15) ;

13.setBounds(40, 180, 100, 15) ;

t1.setBounds(200, 100, 100, 15) ;

t2.setBounds(200, 140, 100, 15) ;

t3.setBounds (200, 180, 100, 15) ;

b1.setBounds (50, 250, 45, 15) ;

b2.setBounds (110, 250, 45, 15) ;

b3.setBounds (170, 250, 45, 15) ;

b4.setBounds (230, 250, 45, 15) ;

b5.setBounds (290, 250, 45, 15) ;

// Adding components to the frame

f1.add(l1);

f1.add(l2);

f1.add(l3);

f1.add(t1);

f1.add(t2);

f1.add(t3);

f1.add(b1);

f1.add(b2);

f1.add(b3);

f1.add(b4);

f1.add(b5);

b1.addActionListener (this);

b2.addActionListener (this);

b3.addActionListener (this);

b4.addActionListener (this);

b5.addActionListener (this);

f1.setLayout (null) ;

f1.setVisib1e (true) ;

f1.setSize(400, 350);

}

public void actionPerformed (ActionEvent e) {

int n1 = Integer.parseInt (t1.get Text ( ) ) ;

int n2 = Integer.parseInt (t2.get Text ( ) ) ;

if (e.getSource() == b1)

{

t3.setText (String. valueOf (n1 + n2) ) ;

}

if (e.getSource() ==b2)

{

t3.setText (String. valueOf (n1 - n2) ) ;

}

if (e.getSource()==b3)

{

t3.setText (String. valueOf (n1 \* n2) ) ;

}

if (e.getSource()==b4)

{

t3.setText (String. valueOf (n1 / n2) ) ;

}

if (e.getSource()==b5)

{

System.exit(0);

}

public static void main(String arg[])

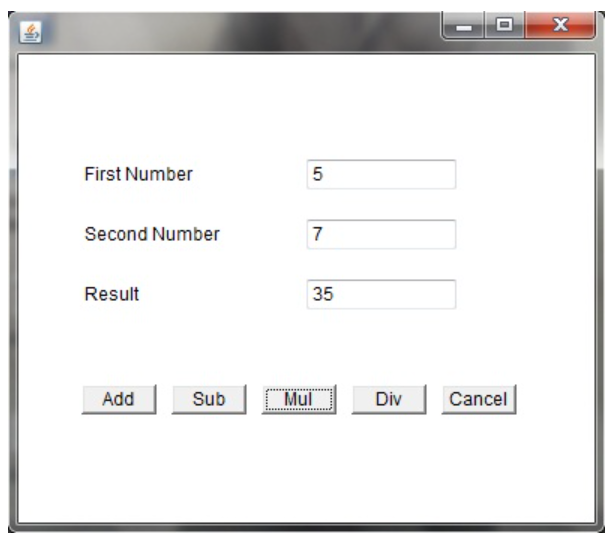
{

new simplecalculator() ;

}

}

Sample Output



**5 b. Write a program to connect Database using JDBC Driver**

**Algorithm for the SimpleDBConnectionDemo Program:**

**Step 1: Start**

**Step 2: Initialize Variables**:

* + Define the **JDBC Driver** class: "com.mysql.jdbc.Driver".
  + Set the **JDBC URL**: "jdbc:mysql://localhost:3306/test".
  + Set the **Username**: "root".
  + Set the **Password**: "" (empty password in this case).

**Step 3: Try to Connect to the Database**:

* + **Step 3.1**: Load the **MySQL JDBC Driver** using Class.forName("com.mysql.jdbc.Driver").
    - If the driver is not found, throw a **ClassNotFoundException**.
  + **Step 3.2**: Establish a connection using DriverManager.getConnection(JDBC\_URL, USERNAME, PASSWORD).
    - If the connection fails, throw an **SQLException**.

**Step 4: Print Success Message**:

* + If the connection is successful, print "Connection Established" along with the connection object details.

**Step 5: Close the Database Connection**:

* + Close the connection using connection.close().
  + Print "Connection Terminated" after successfully closing the connection.

**Step 6: Handle Exceptions**:

* + **ClassNotFoundException**: If the JDBC driver is not found, print the error stack trace.
  + **SQLException**: If there is a problem establishing the connection (e.g., invalid credentials or database is down), print the error stack trace.

**Step 7: End**

**Program:**

import java.sql.\*;

public class DBConnect {

public static void main(String [] args) {

try {

Class.forName("com.mysql.jdbc.Driver"); // Load the Driver

// Connect to the database.

// Protocol

// IPAddress

// Port Number

// Database (Collection of Tables)

Connection connection =

DriverManager.getConnection("jdbc:mysql://localhost:3306/test",

"root","");

System.out.println("Connection Established" +

connection);

connection.close();

System.out.println("Connection Terminated");

} catch (ClassNotFoundException | SQLException e) {

e.printStackTrace();

}

}

}

**Sample Output:**

Connection Established com.mysql.jdbc.JDBC4Connection@1c53fd30

Connection Terminated