CSE1007-JAVA PROGRAMMING-LAB
EXERCISE-05

Name: Kulvir Singh
Reg. No.: 19BCE2074

Question 1

JavaFX Application

Part a)

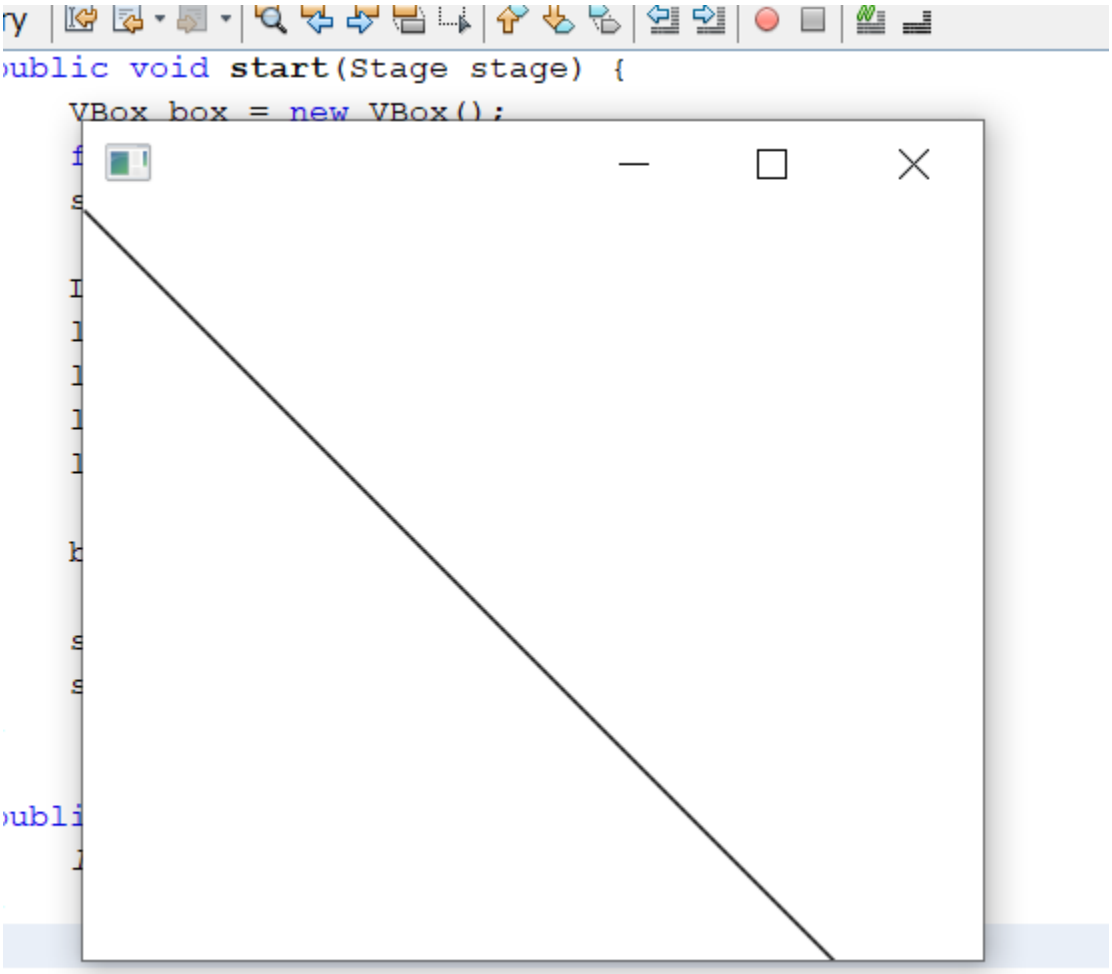
Aim:
A straight line with the given starting and ending coordinates

Code:

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.layout.VBox;
import javafx.scene.shape.Line;
import javafx.stage.Stage;
public class JavaFXApplication1 extends Application
{

   @Override
   public void start(Stage stage) {
      VBox box = new VBox();
      final Scene scene = new Scene(box,300, 250);
      scene.setFill(null);

      Line line = new Line();
      line.setStartX(0.0f);
      line.setStartY(0.0f);
      line.setEndX(1000.0f);
      line.setEndY(1000.0f);

      box.getChildren().add(line);

      stage.setScene(scene);
      stage.show();
   }

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

```
        launch(args);
    }
}
```

Output Screenshot:



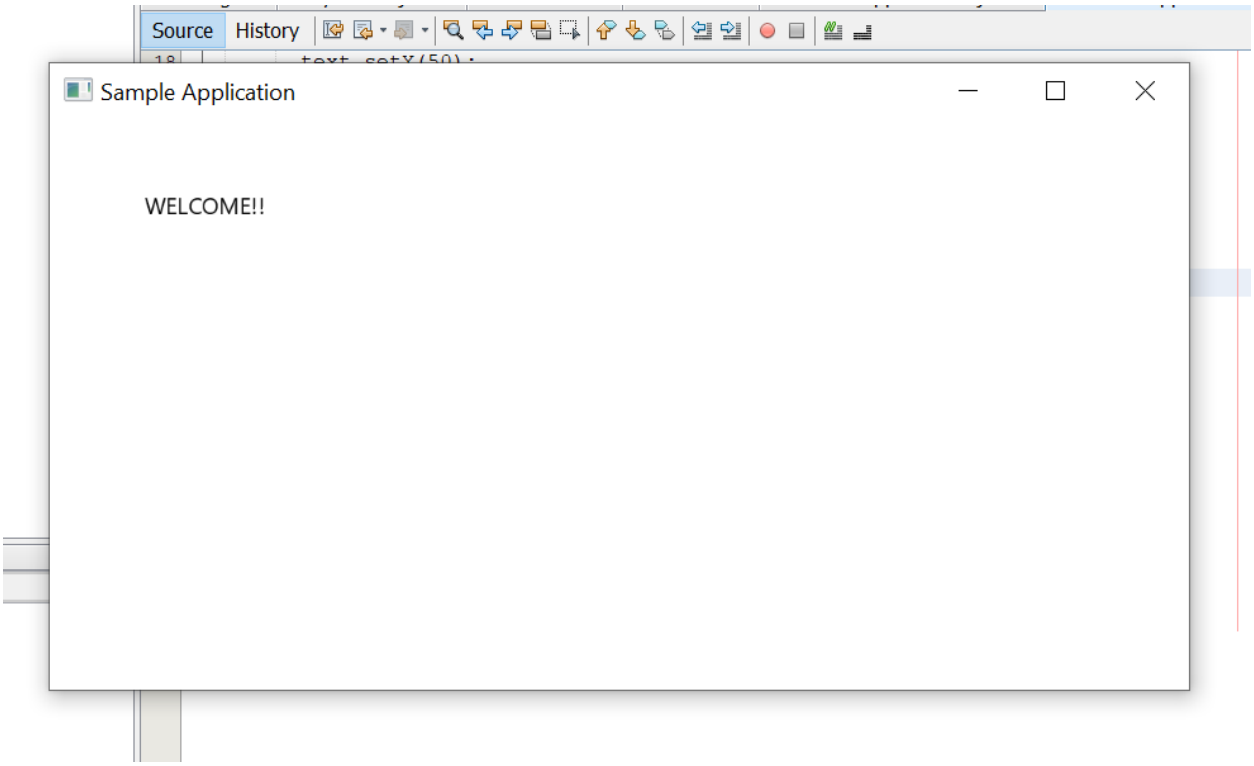
Part b)

Aim:
A welcome text in the scene.

Code:

```
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.stage.Stage;
import javafx.scene.text.Text;

public class JavaFXApplication2 extends Application {
  @Override
  public void start(Stage stage) {
    //Creating a Text object
    Text text = new Text();

    //Setting the text to be added.
    text.setText("WELCOME!!");

    //setting the position of the text
    text.setX(50);
    text.setY(50);

    //Creating a Group object
    Group root = new Group(text);

    //Creating a scene object
    Scene scene = new Scene(root, 600, 300);

    //Setting title to the Stage
    stage.setTitle("Sample Application");

    //Adding scene to the stage
    stage.setScene(scene);

    //Displaying the contents of the stage
    stage.show();
  }
  public static void main(String args[]){
    launch(args);
  }
}
```

Output Screenshot:



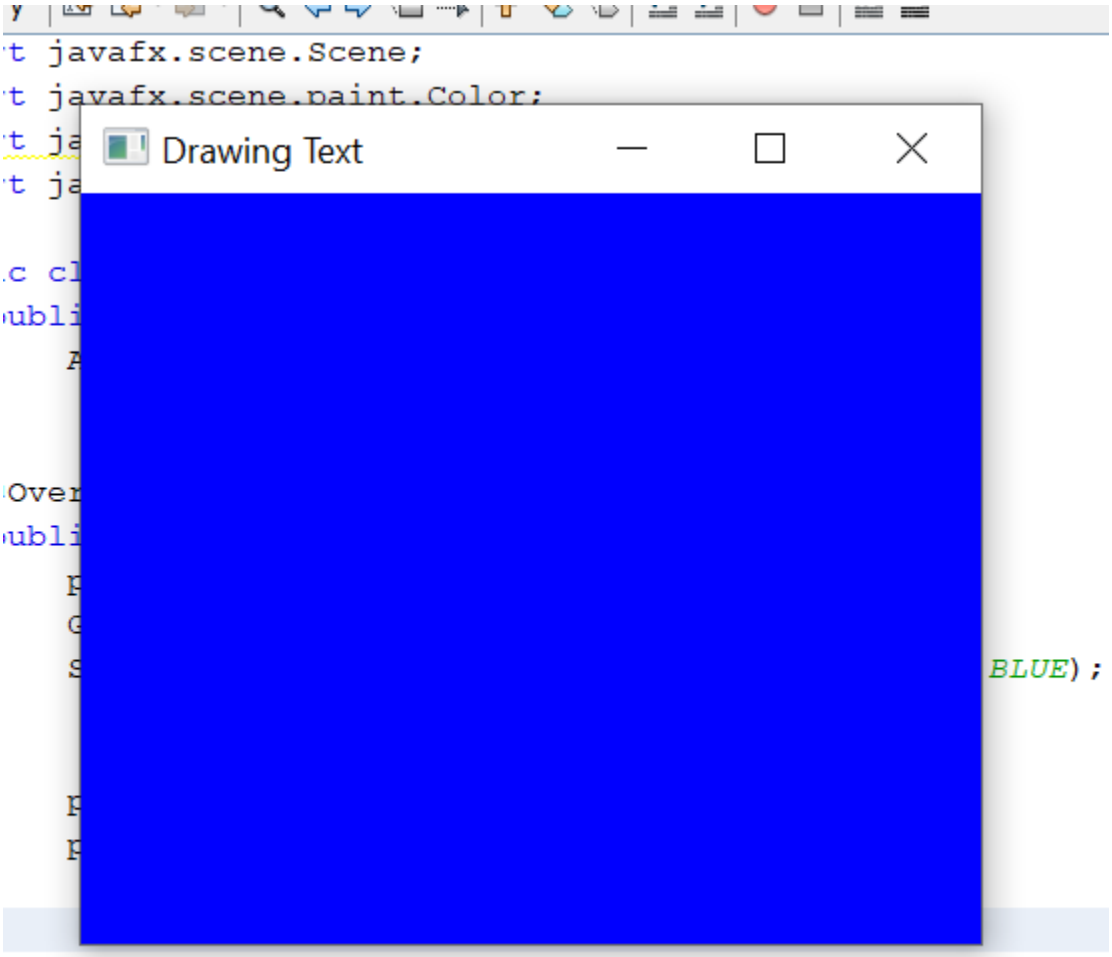
Part c)

Aim:
a blue background colour.

Code:

```
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.paint.Color;
import javafx.scene.text.Text;
import javafx.stage.Stage;

public class JavaFXApplication3 extends Application {
   public static void main(String[] args) {
      Application.launch(args);
   }

   @Override
   public void start(Stage primaryStage) {
      primaryStage.setTitle("Drawing Text");
      Group root = new Group();
      Scene scene = new Scene(root, 300, 250, Color.BLUE);


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

Output Screenshot:



Question 2

Aim:

Demonstrate the Java Database Connectivity (JDBC) by connecting a java application with a database. Perform basic operations of database.

Code:

```
import java.sql.*;
public class Demo {
   public static void main(String[] args) throws SQLException {
      Connection myConn = null;
      Statement myStmt = null;
      ResultSet myRs = null;
      String user = "root";
      String pass = "password";

      try {
          myConn = DriverManager.getConnection("jdbc:mysql://localhost:3306/jdbc", user, pass);
         System.out.println("inserting values into database");
         myStmt = myConn.createStatement();
         int updt;
         updt = myStmt.executeUpdate("insert into entries values(\"cristiano\",35)");
      } catch (Exception exc) {
         exc.printStackTrace();
      } finally {
         if (myRs != null) {
            myRs.close();
         }
         if (myStmt != null) {
            myStmt.close();
         }
         if (myConn != null) {
            myConn.close();
         }
```

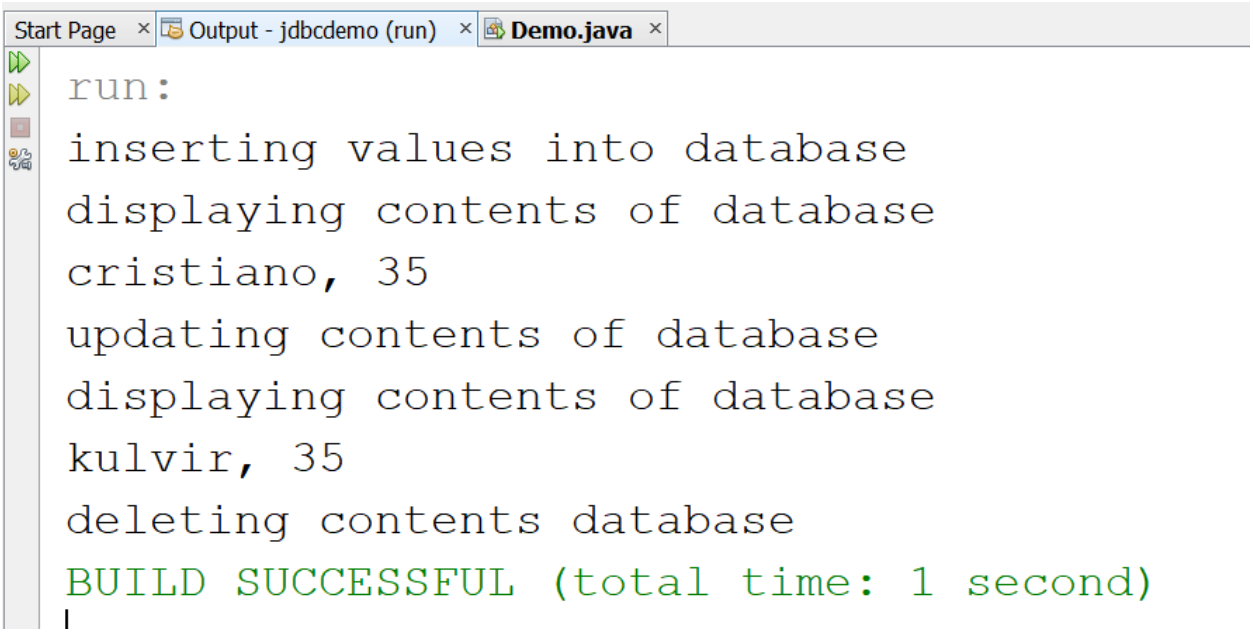
```
}
try {
   myConn = DriverManager.getConnection("jdbc:mysql://localhost:3306/jdbc", user, pass);
   myStmt = myConn.createStatement();
   System.out.println("displaying contents of database");
   myRs = myStmt.executeQuery("select * from entries");
   while (myRs.next()) {
      //System.out.println("Name = "myRs.getString("name")+" , Age = "+myRs.getString("age"));
      System.out.println(myRs.getString("name") + ", " + myRs.getString("age"));
   }

} catch (Exception exc) {
   exc.printStackTrace();
} finally {
   if (myRs != null) {
      myRs.close();
   }

   if (myStmt != null) {
      myStmt.close();
   }

   if (myConn != null) {
      myConn.close();
   }
}
try {
   myConn = DriverManager.getConnection("jdbc:mysql://localhost:3306/jdbc", user, pass);
   System.out.println("updating contents of database");
```

```
            myStmt = myConn.createStatement();
            int updt;
            updt = myStmt.executeUpdate("update entries set name = \"kulvir\" where name =
\"cristiano\"");
        } catch (Exception exc) {
            exc.printStackTrace();
        } finally {
            if (myRs != null) {
                myRs.close();
            }

            if (myStmt != null) {
                myStmt.close();
            }

            if (myConn != null) {
                myConn.close();
            }
        }
        try {
            myConn = DriverManager.getConnection("jdbc:mysql://localhost:3306/jdbc", user, pass);
            myStmt = myConn.createStatement();
            System.out.println("displaying contents of database");
            myRs = myStmt.executeQuery("select * from entries");
            while (myRs.next()) {
                System.out.println(myRs.getString("name") + ", " + myRs.getString("age"));
            }

        } catch (Exception exc) {
```

```
    exc.printStackTrace();
} finally {
    if (myRs != null) {
        myRs.close();
    }

    if (myStmt != null) {
        myStmt.close();
    }

    if (myConn != null) {
        myConn.close();
    }
}
try {
    myConn = DriverManager.getConnection("jdbc:mysql://localhost:3306/jdbc", user, pass);
    myStmt = myConn.createStatement();
    System.out.println("deleting contents database");
    int updt;
    updt = myStmt.executeUpdate("truncate table entries");
} catch (Exception exc) {
    exc.printStackTrace();
} finally {
    if (myRs != null) {
        myRs.close();
    }
    if (myStmt != null) {
        myStmt.close();
    }
```

```
        if (myConn != null) {
            myConn.close();
        }
    }
  }
}
```

Output Screenshot:



Question 3

Aim:

Implement a java servlet by configuring a webserver (apache tomcat) to demonstrate the request-response sequence of server side programming.

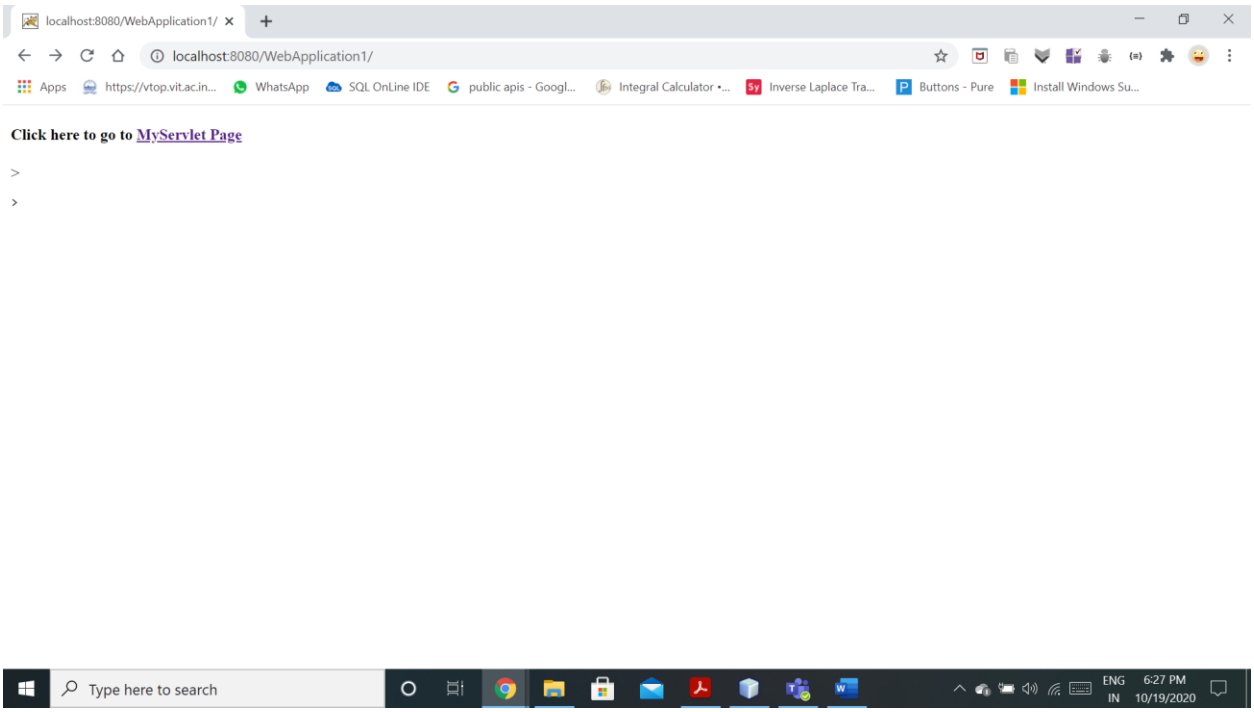
Code:

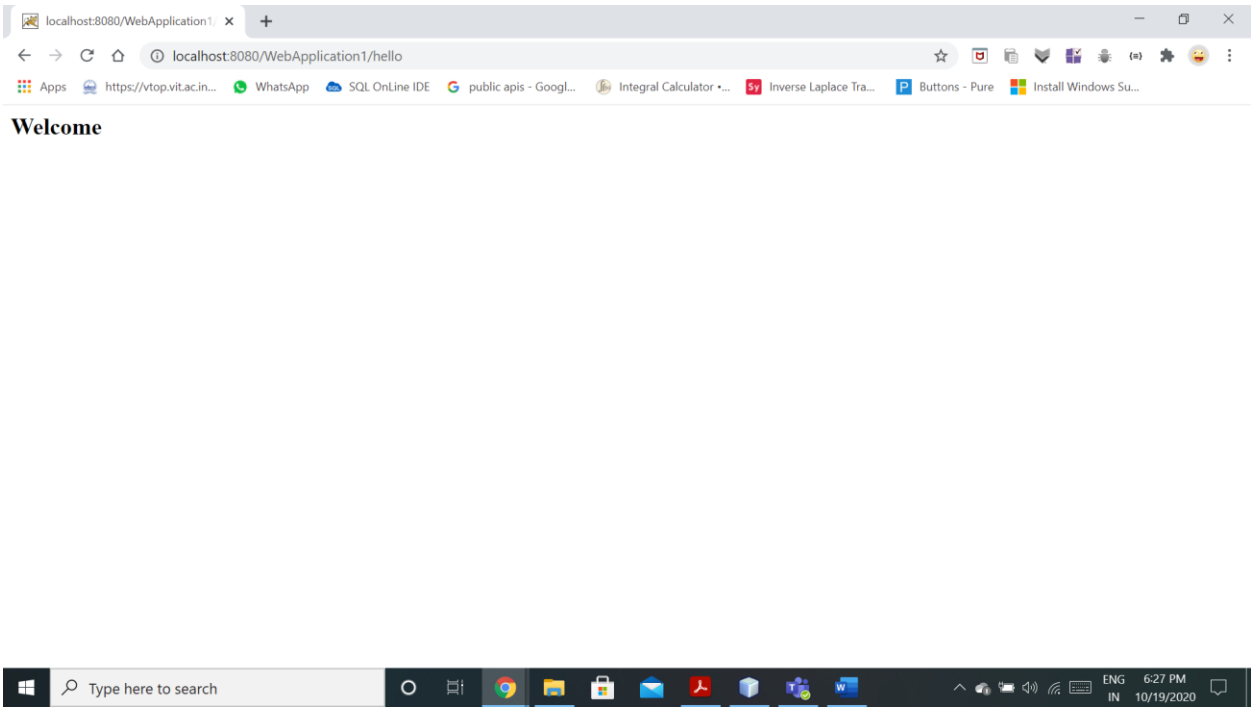
```
import java.io.IOException;
import java.io.PrintWriter;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class MyServlet extends HttpServlet {
   protected void doGet(HttpServletRequest request,HttpServletResponse response)
      throws ServletException, IOException{
      response.setContentType("text/html;charset=UTF-8");
      PrintWriter out = response.getWriter();
      try{
          out.println("<h2>Welcome</h2>");
      }
      finally{
      out.close();}

   }

   }
```

HTML CODE

```
<!DOCTYPE html>
<html>
   <head>
      <title></title>>
   </head>
   <body>
      <h4>Click here to go to <a href ="hello">MyServlet Page</a></h4>
   </body>>
</html>
```

Output Screenshot:





TOMCAT SERVER EXECUTION IN NETBEANS SCREENSHOT: