UNIVERSITI MALAYSIA TERENGGANU

CSF3123 DATABASE (K2)

BACHELOR OF COMPUTER SCIENCE (MOBILE COMPUTING) WITH HONORS

LAB 1

SEMESTER II 2023/2024

Prepared for:

SIR MOHD ARIZAL SHAMSIL BIN MAT RIFIN

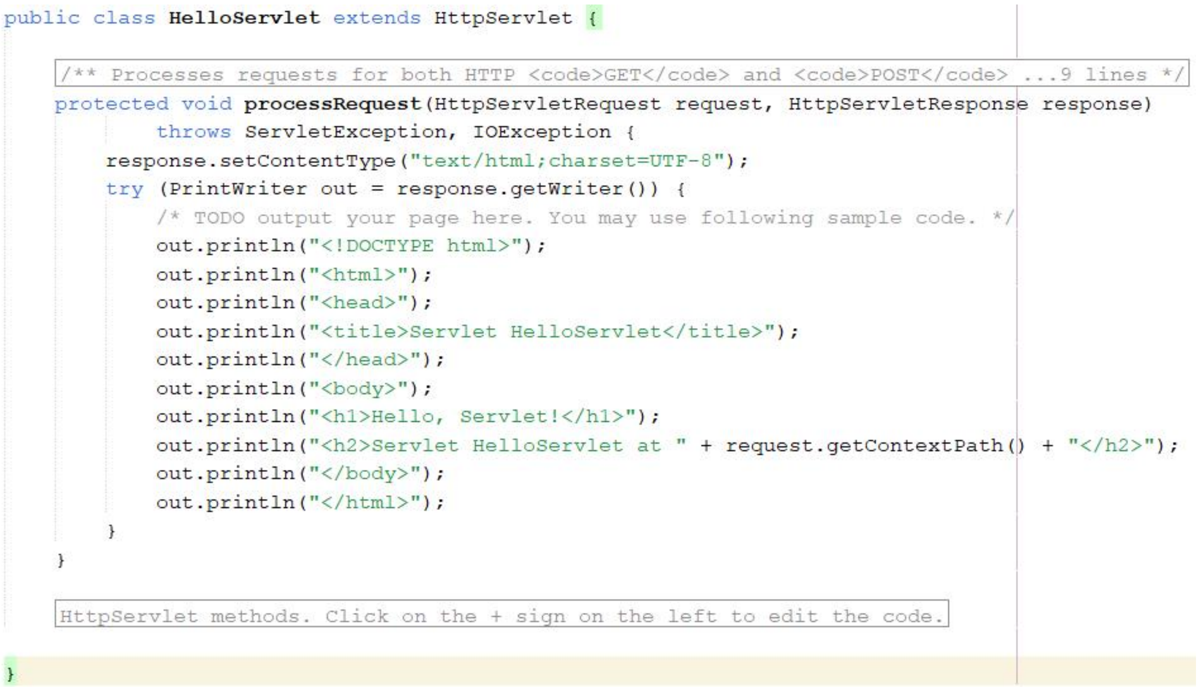
Prepared by:

MUHAMMAD ZAHIER BIN RAZMI

(S67943)

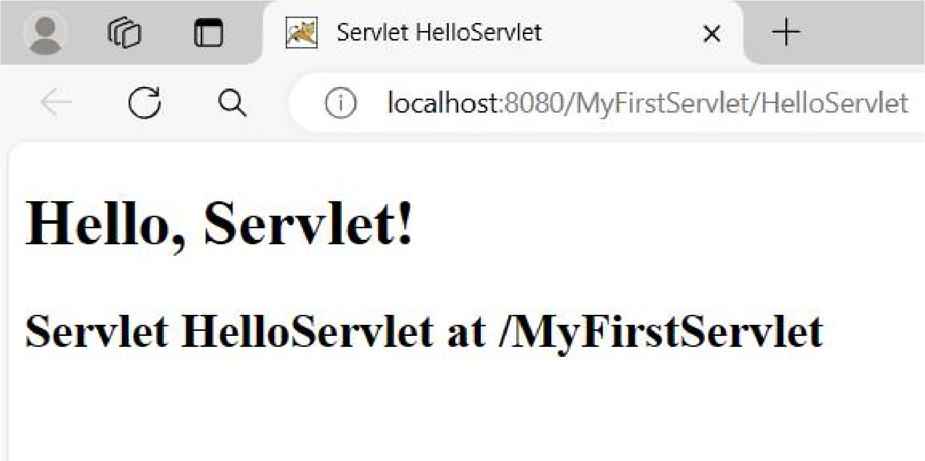
Task 5: Linking Netbeans to Apache Tomcat and Writing a Simple Java Servlet

Coding:



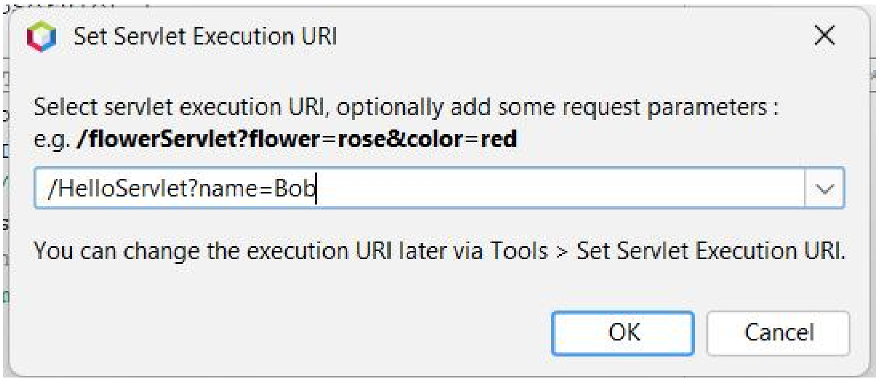
```
public class HelloServlet extends HttpServlet {

    /** Processes requests for both HTTP <code>GET</code> and <code>POST</code> ...9 lines */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use following sample code. */
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet HelloServlet</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Hello, Servlet!</h1>");
            out.println("<h2>Servlet HelloServlet at " + request.getContextPath() + "</h2>");
            out.println("</body>");
            out.println("</html>");
        }
    }

    HttpServlet methods. Click on the + sign on the left to edit the code.

}
```

Output



Hello, Servlet!

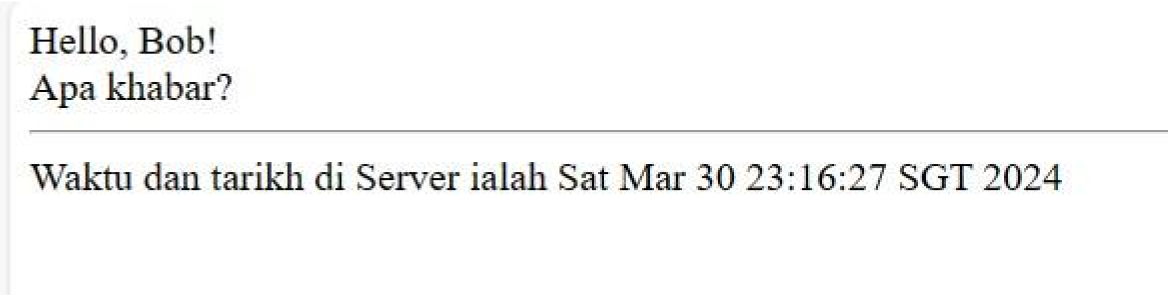
Servlet HelloServlet at /MyFirstServlet



Select servlet execution URI, optionally add some request parameters :
e.g. **/flowerServlet?flower=rose&color=red**

/HelloServlet?name=Bob

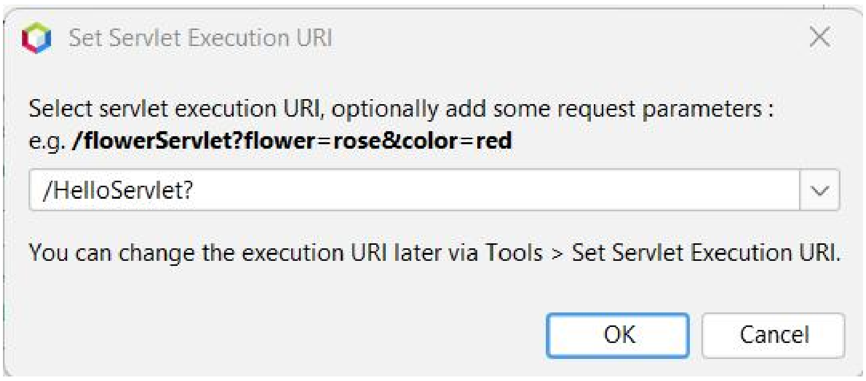
You can change the execution URI later via Tools > Set Servlet Execution URI.



Hello, Bob!
Apa khabar?

Waktu dan tarikh di Server ialah Sat Mar 30 23:16:27 SGT 2024

Without parameter

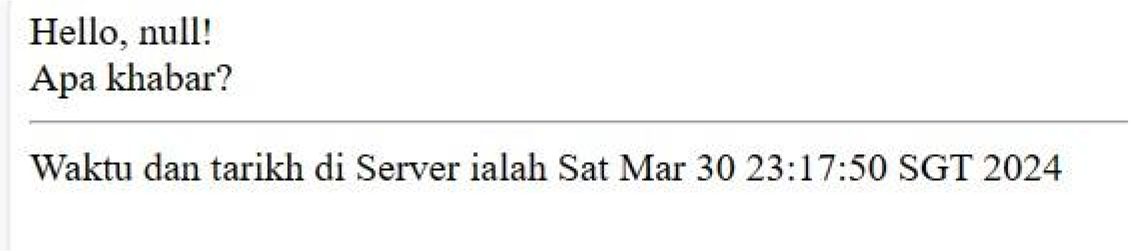


Set Servlet Execution URI

Select servlet execution URI, optionally add some request parameters :
e.g. **/flowerServlet?flower=rose&color=red**

/HelloServlet?

You can change the execution URI later via Tools > Set Servlet Execution URI.



Hello, null!
Apa khabar?

Waktu dan tarikh di Server ialah Sat Mar 30 23:17:50 SGT 2024

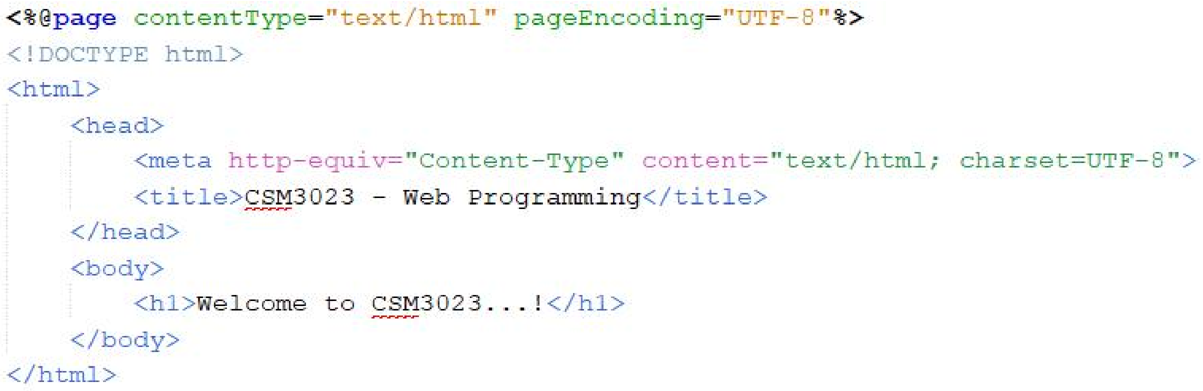
- Reflection:

  It is the same output as can be seen in Step 18. Why?

  It is the same output because the same method processRequest() was recalled.

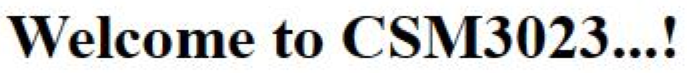
Task 6 : Writing a Simple JSP Program

Coding:



```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>CSM3023 - Web Programming</title>
    </head>
    <body>
        <h1>Welcome to CSM3023...!</h1>
    </body>
</html>
```

Output:



Welcome to CSM3023...!

- Reflection:

1. What have you learned from this exercise?

   From this exercise I learnt how to set up a JSP program in NetBeans using Apache Tomcat server.

2. Explain the general concept of how the JSP's file work?

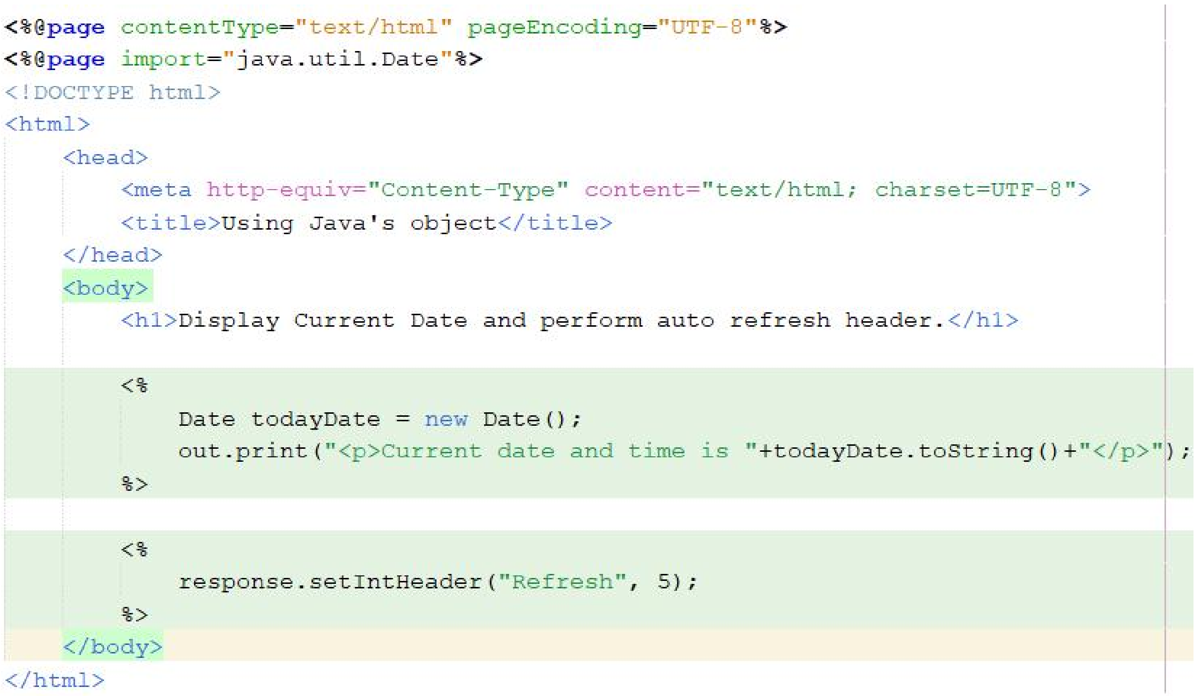
   JSP files, or Java Server Pages, are the secret sauce behind dynamic web content. These files mix regular HTML with special JSP tags. When a JSP file is requested, the server doesn't just send it raw. Instead, it translates the JSP into a special Java program called a servlet. This servlet can interact with databases, other programs, or anything else on the server. Finally, the servlet generates a standard HTML page that gets sent back to the user's browser. This way, JSP pages can create different content for each user, based on their requests or information stored on the server.

3. Based on your observation of the previous tasks (Task 3 and Task 4), what are the differences you can find between servlet and JSP?

   Servlets and JSPs are both the fundamental blocks building a web application, but they have different functions. Servlets can be equated to backstage workers who are written in a pure language (Java) and are responsible for the heavy lifting tasks such as data processing and communication with databases. The distinct aspect of JSPs is the emphasis on the presentation. They blend HTML with JSP tags and are of easy writing than classic versions as they develop don HTML familiarity. JSPs are turned into servlets before executing, which JSPs do not have the other servlets' ability to deal with non-HTTP requests or to directly override core works. In general, a servlet processes the application logic, and a JSP page then is responsible for how the user sees it.

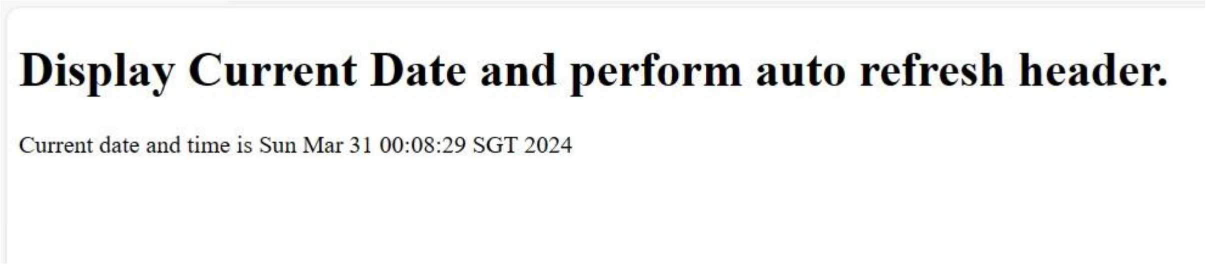
Task 7 : Use Java Reference Datatype/Class Wrapper in JSP

Coding:



```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="java.util.Date"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Using Java's object</title>
    </head>
    <body>
        <h1>Display Current Date and perform auto refresh header.</h1>

        <%
            Date todayDate = new Date();
            out.print("<p>Current date and time is "+todayDate.toString()+"</p>");
        %>

        <%
            response.setIntHeader("Refresh", 5);
        %>
    </body>
</html>
```

Output:



Display Current Date and perform auto refresh header.

Current date and time is Sun Mar 31 00:08:29 SGT 2024

- Reflection:

1. What have you learnt from this exercise?

   From this exercise I learnt how to use the <%%> symbol which is scriptlet tag used to embed Java code within the HTML content of a JSP file. We can insert Java code in HTML by using scriptlet tag.

2. What is Java Scriptlet?

   A Java Scriptlet is a block of Java code slipped directly between speech tags in a JSP page. These become `<% %>` encoded. With its use of scriptlets, you are able to inser dynamic logic and get swing to the server-side resources that can take place right within your JSP file. This enables you to accomplish such as assignments of the variables to the received data, open the databases, or do everything with calculations. Though mini-scripts are helpful, they can complicate code base and deteriorate it dealing with security.

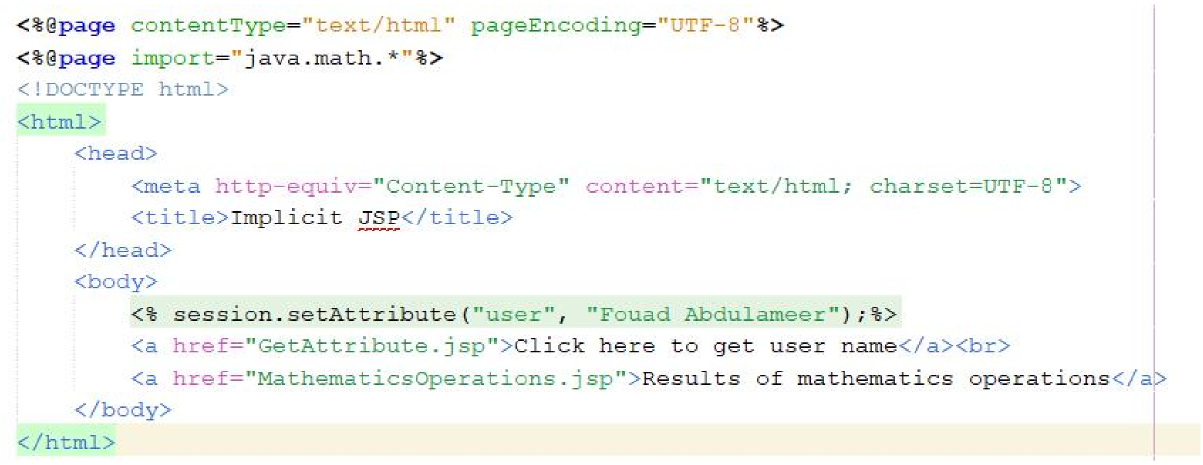
3. How to use Java code in your JSP's page?

   We can use scriptlets, expressions, declarations, and directives to accomplish it. Chunks of Java code which is enclosed in `<% %>` tags are called scriptlets. The scriptlets allow operations like data processing, decision making, and the looping. Things in angels brackets will be used to evaluate and output Java expressions being directly rendered into the HTML response. Declarations, denoted by `<%! <code></code> are utilized to define both variables and methods as well as classes that are needed throughout the JSP page.

Task 8 : Using JSP Implicit object in JSP page

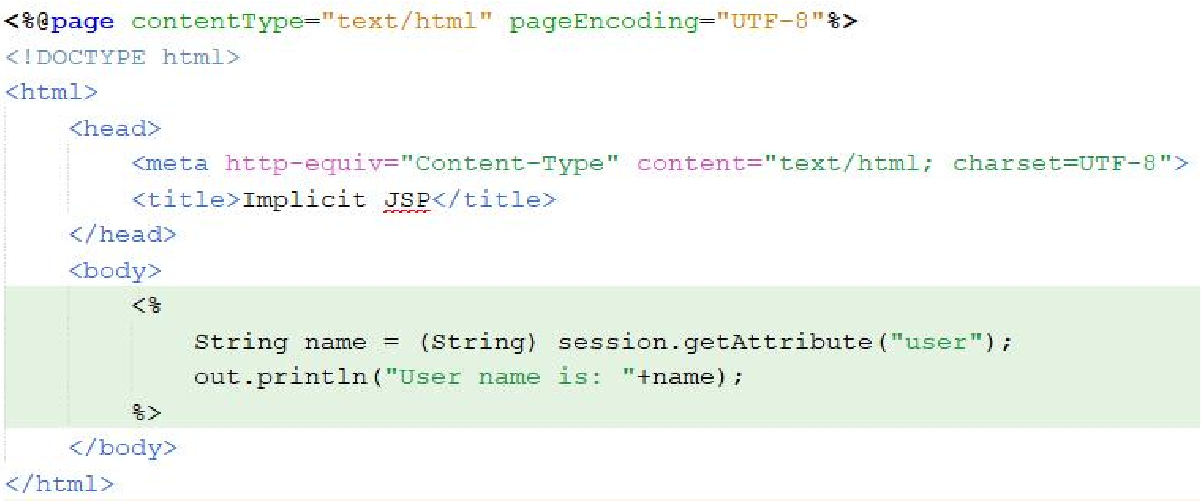
Coding:

AttributeIsSet



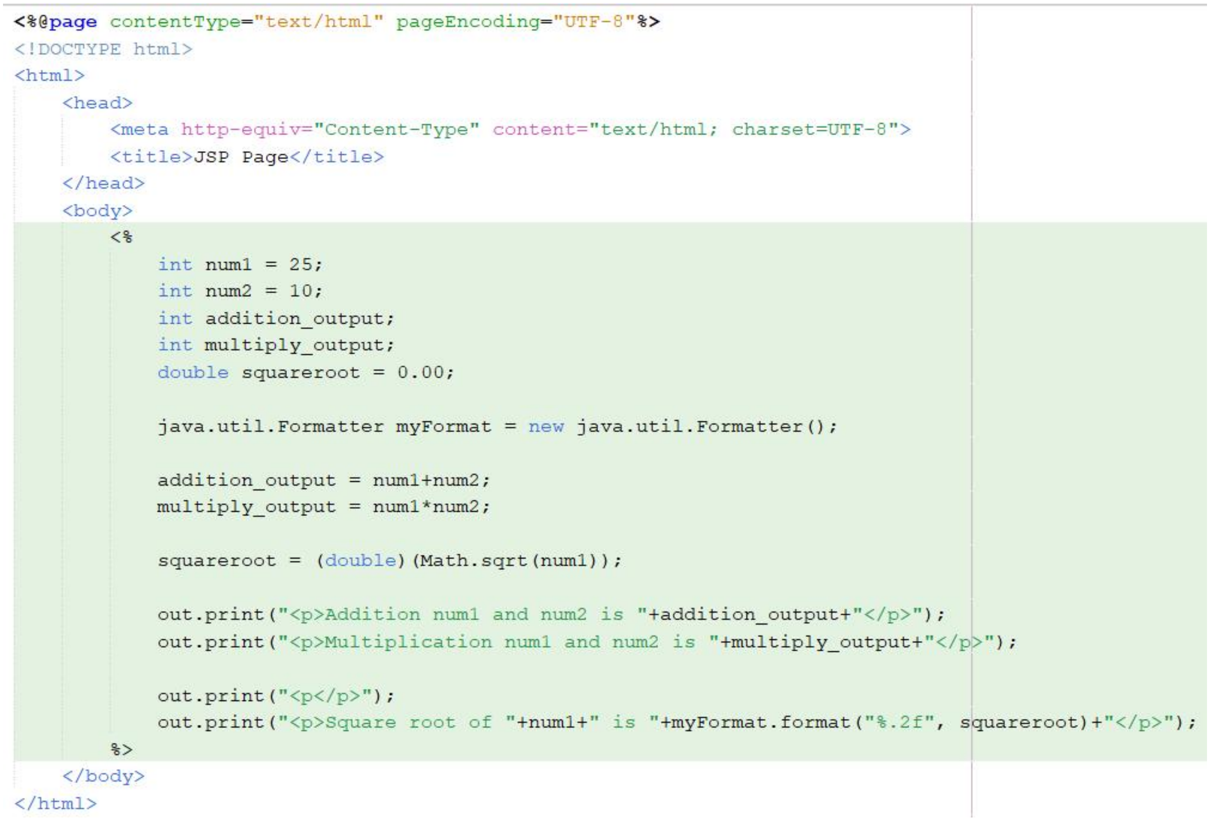
```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="java.math.*"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Implicit JSP</title>
    </head>
    <body>
        <% session.setAttribute("user", "Fouad Abdulameer");%>
        <a href="GetAttribute.jsp">Click here to get user name</a><br>
        <a href="MathematicsOperations.jsp">Results of mathematics operations</a>
    </body>
</html>
```

GetAttribute



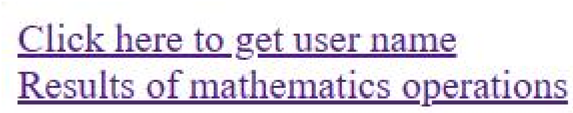
```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Implicit JSP</title>
    </head>
    <body>
        <%
            String name = (String) session.getAttribute("user");
            out.println("User name is: "+name);
        %>
    </body>
</html>
```

MathematicsOperations



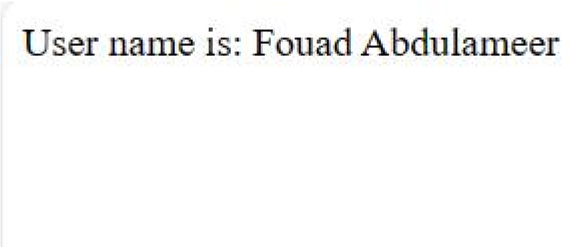
```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <%
            int num1 = 25;
            int num2 = 10;
            int addition_output;
            int multiply_output;
            double squareroot = 0.00;

            java.util.Formatter myFormat = new java.util.Formatter();

            addition_output = num1+num2;
            multiply_output = num1*num2;

            squareroot = (double)(Math.sqrt(num1));

            out.print("<p>Addition num1 and num2 is "+addition_output+"</p>");
            out.print("<p>Multiplication num1 and num2 is "+multiply_output+"</p>");

            out.print("<p></p>");
            out.print("<p>Square root of "+num1+" is "+myFormat.format("%.2f", squareroot)+"</p>");
        %>
    </body>
</html>
```

Output:

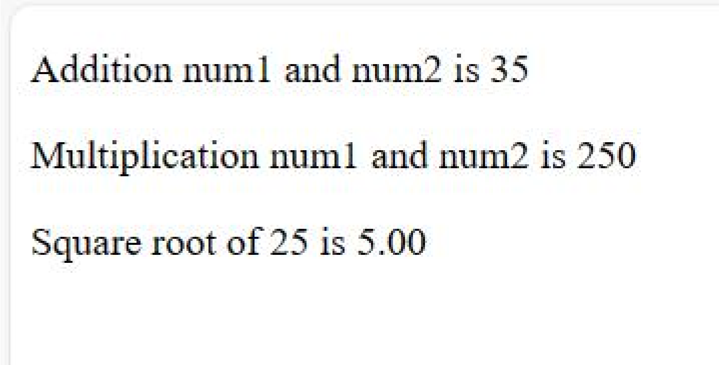


Click here to get user name

Results of mathematics operations



User name is: Fouad Abdulameer



Addition num1 and num2 is 35

Multiplication num1 and num2 is 250

Square root of 25 is 5.00

- Reflection:

1. How do you want to submit specific information from one form to next form?

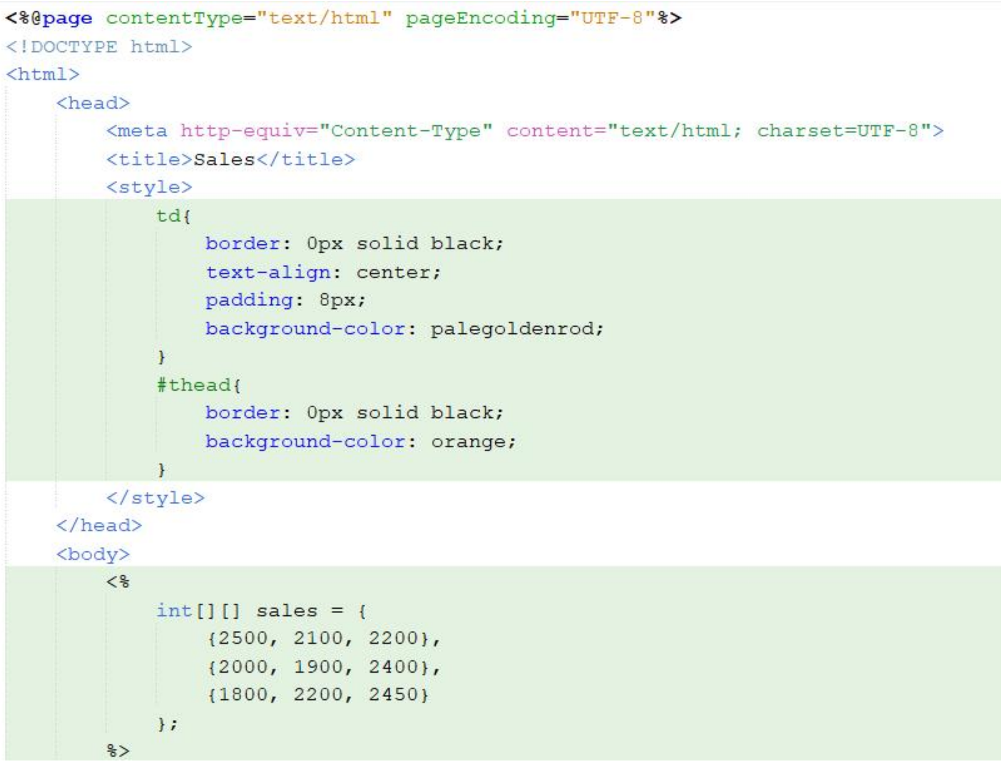
   By using href attributes. It is because href attribute specifies the URL of the page that the link goes to or the page next destination, like hyperlink.

2. What happened if the field name you specify in request.getParameter("field_name") in the second page is different from the field name you defined in the first page?

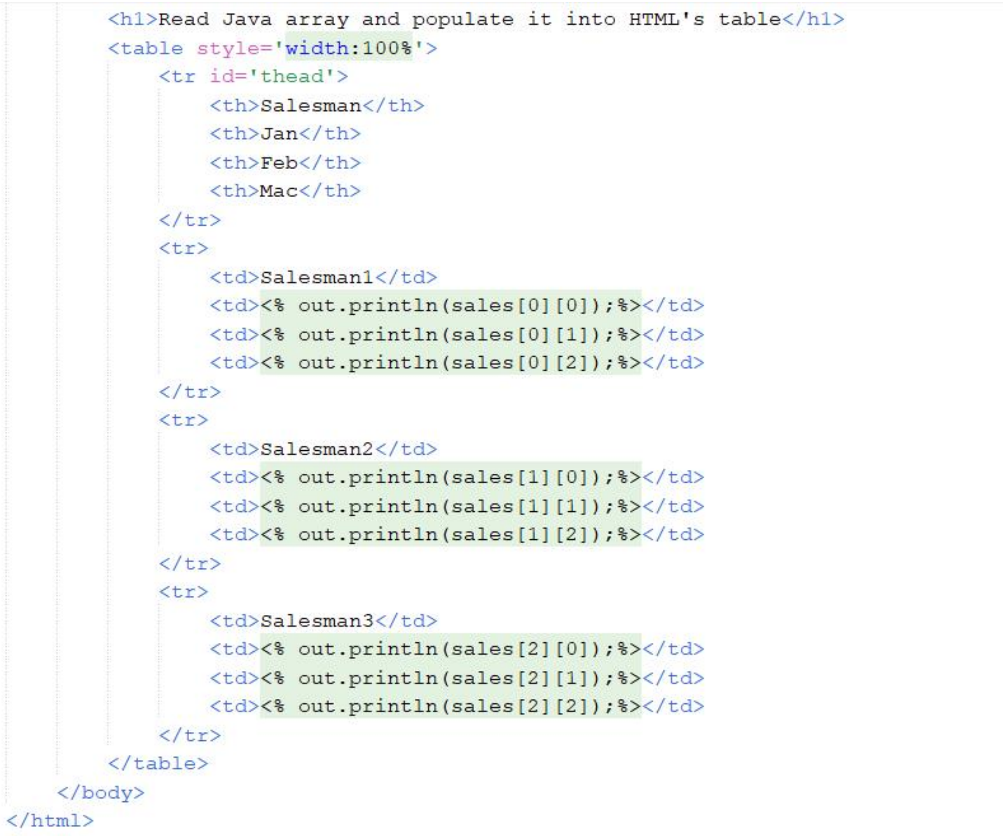
   It will show null.

Task 9 : Populate Array values into HTML’s Table

Coding:

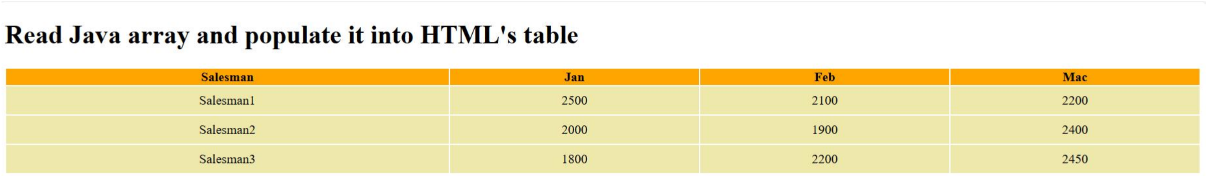


```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Sales</title>
        <style>
            td{
                border: 0px solid black;
                text-align: center;
                padding: 8px;
                background-color: palegoldenrod;
            }
            #thead{
                border: 0px solid black;
                background-color: orange;
            }
        </style>
    </head>
    <body>
        <%
            int[][] sales = {
                {2500, 2100, 2200},
                {2000, 1900, 2400},
                {1800, 2200, 2450}
            };
        %>
```



```
        <h1>Read Java array and populate it into HTML's table</h1>
        <table style='width:100%'>
            <tr id='thead'>
                <th>Salesman</th>
                <th>Jan</th>
                <th>Feb</th>
                <th>Mac</th>
            </tr>
            <tr>
                <td>Salesman1</td>
                <td><% out.println(sales[0][0]);%></td>
                <td><% out.println(sales[0][1]);%></td>
                <td><% out.println(sales[0][2]);%></td>
            </tr>
            <tr>
                <td>Salesman2</td>
                <td><% out.println(sales[1][0]);%></td>
                <td><% out.println(sales[1][1]);%></td>
                <td><% out.println(sales[1][2]);%></td>
            </tr>
            <tr>
                <td>Salesman3</td>
                <td><% out.println(sales[2][0]);%></td>
                <td><% out.println(sales[2][1]);%></td>
                <td><% out.println(sales[2][2]);%></td>
            </tr>
        </table>
    </body>
</html>
```

Output:



Read Java array and populate it into HTML's table

Salesman	Jan	Feb	Mac
Salesman1	2500	2100	2200
Salesman2	2000	1900	2400
Salesman3	1800	2200	2450

- Reflection:

1. Write a sample syntax to declare 2D Java array.

```
datatype[][] arrayName = new dataType[row][colums];
```

2. Define a sequence of steps on how you accomplish Task 7.
   - Create a new JSP file and name it as useJavaObject.
   - Import java.util.Date.
   - Create Date object and display the current date and time.
   - Use setIntHeader() to trigger refresh function and set the autoload time to 5 seconds.
   - Compile and run the program.

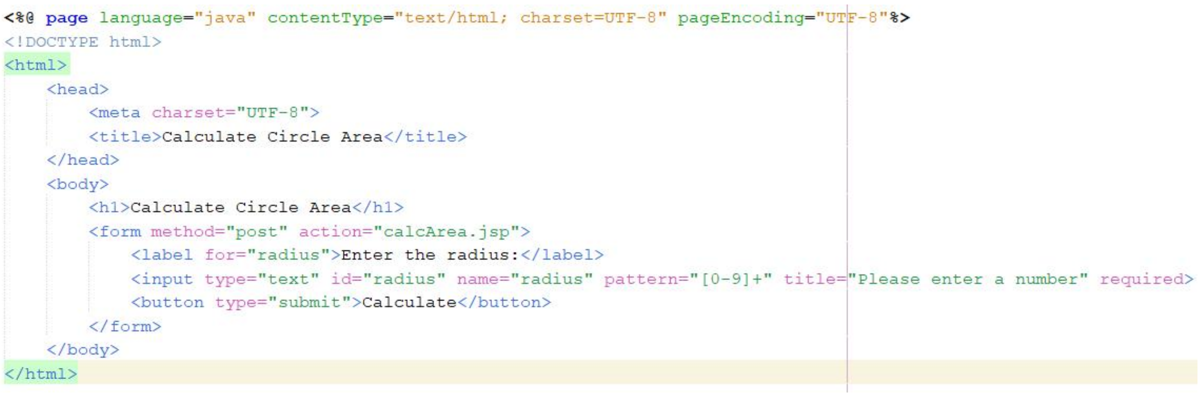
3. What is the difference between HTML's page and JSP's page?

HTML standing for Hypertext Markup Language is a kind of webpage where the content and structure are static, which is often perfect or fast to display regular content on the web. Furthermore, the improvement of JSP (JavaServer Pages) pages is dynamic web pages, which can contain both HTML as well as Java program. In contrast to JSP pages that embed java code inside of an HTML markup, HTML pages allow only a static presentation of content that can be generated using HTML code alone. JSP pages firstly go through the run-time processing from one of the most widely used server-side engines, in which the Java code is interpreted and then executed to create the dynamic HTML content based on the results of the calculation. This enables the development of interactive and data-driven web apps with the capability to fetch data from different sources and tailor the content based on the requested information. Apart from it, it offers the functionalities that are not restricted to the custom tags, but it provides easier interface with Java servlets as well as other backend technology and helps better separate the presentations and business logic.

Exercise 1

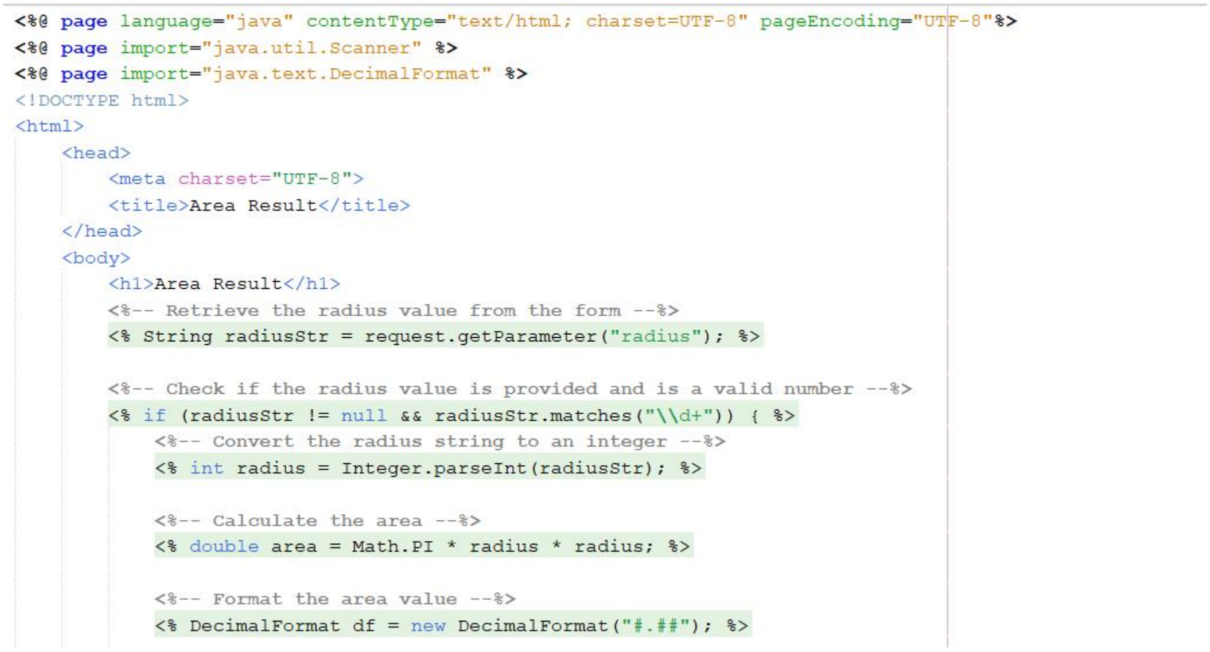
Coding:

getRadius

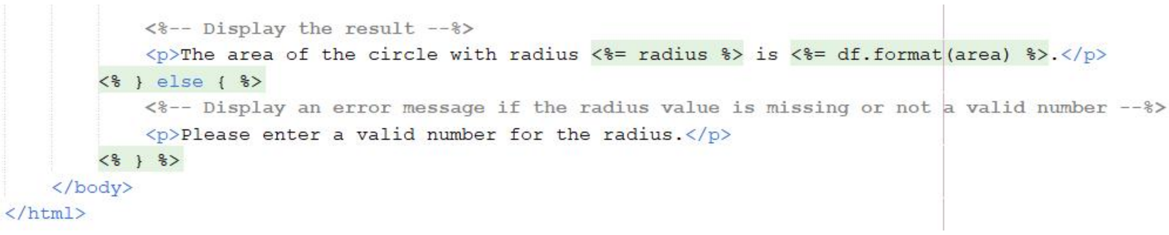


```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Calculate Circle Area</title>
    </head>
    <body>
        <h1>Calculate Circle Area</h1>
        <form method="post" action="calcArea.jsp">
            <label for="radius">Enter the radius:</label>
            <input type="text" id="radius" name="radius" pattern="[0-9]+" title="Please enter a number" required>
            <button type="submit">Calculate</button>
        </form>
    </body>
</html>
```

calcArea

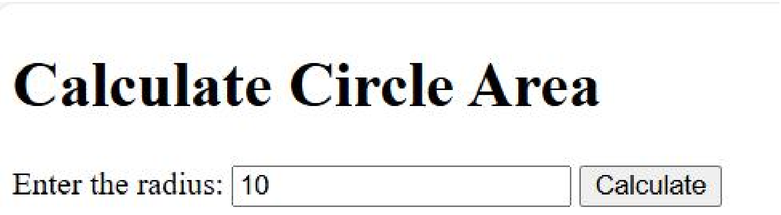


```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ page import="java.util.Scanner" %>
<%@ page import="java.text.DecimalFormat" %>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Area Result</title>
    </head>
    <body>
        <h1>Area Result</h1>
        <%-- Retrieve the radius value from the form --%>
        <% String radiusStr = request.getParameter("radius"); %>

        <%-- Check if the radius value is provided and is a valid number --%>
        <% if (radiusStr != null && radiusStr.matches("\\d+")) { %>
            <%-- Convert the radius string to an integer --%>
            <% int radius = Integer.parseInt(radiusStr); %>

            <%-- Calculate the area --%>
            <% double area = Math.PI * radius * radius; %>

            <%-- Format the area value --%>
            <% DecimalFormat df = new DecimalFormat("#.##"); %>
```



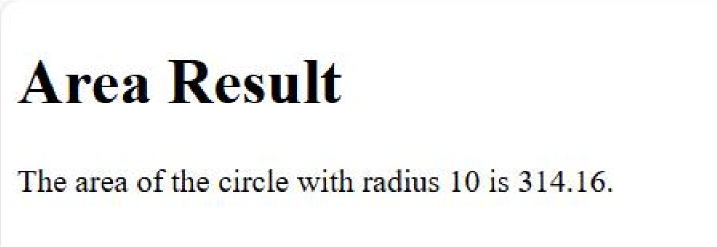
```
            <%-- Display the result --%>
            <p>The area of the circle with radius <%= radius %> is <%= df.format(area) %>.</p>
        <% } else { %>
            <%-- Display an error message if the radius value is missing or not a valid number --%>
            <p>Please enter a valid number for the radius.</p>
        <% } %>
    </body>
</html>
```

Output:



Calculate Circle Area

Enter the radius: 10

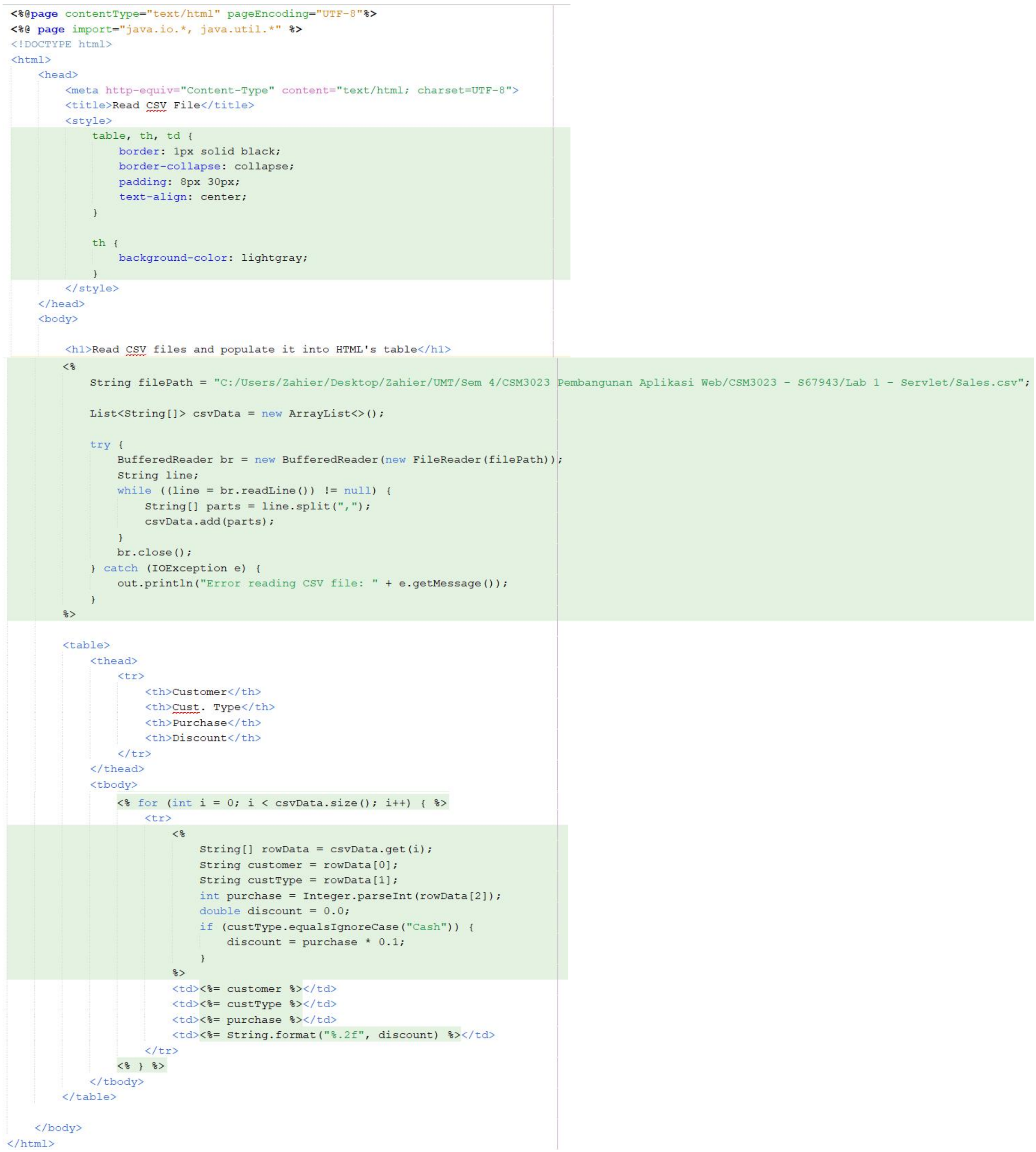


Area Result

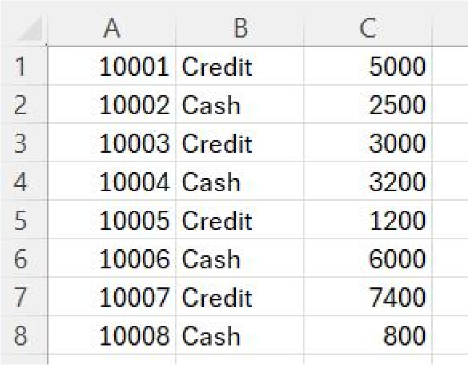
The area of the circle with radius 10 is 314.16.

Exercise 2

Coding:

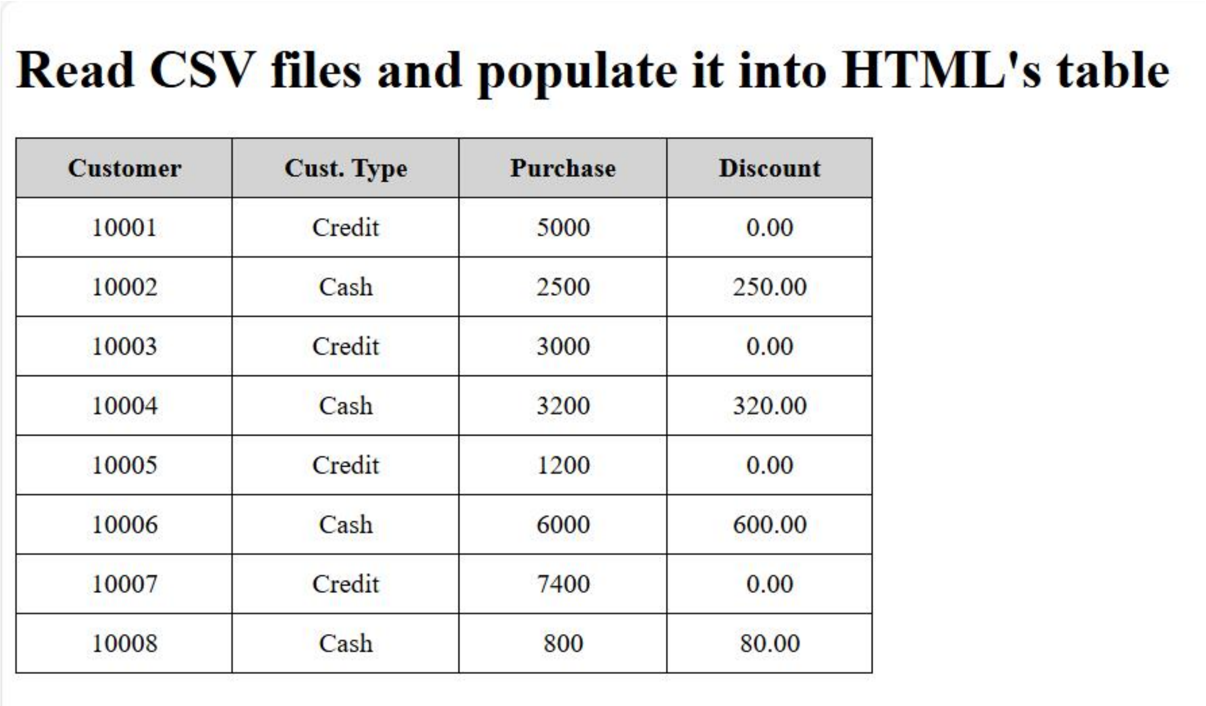


```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ page import="java.io.*, java.util.*" %>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Read CSV File</title>
        <style>
            table, th, td {
                border: 1px solid black;
                border-collapse: collapse;
                padding: 8px 30px;
                text-align: center;
            }

            th {
                background-color: lightgray;
            }
        </style>
    </head>
    <body>

        <h1>Read CSV files and populate it into HTML's table</h1>
        <%
            String filePath = "C:/Users/Zahier/Desktop/Zahier/UMT/Sem 4/CSM3023 Pembangunan Aplikasi Web/CSM3023 - S67943/Lab 1 - Servlet/Sales.csv";

            List<String[]> csvData = new ArrayList<>();

            try {
                BufferedReader br = new BufferedReader(new FileReader(filePath));
                String line;
                while ((line = br.readLine()) != null) {
                    String[] parts = line.split(",");
                    csvData.add(parts);
                }
                br.close();
            } catch (IOException e) {
                out.println("Error reading CSV file: " + e.getMessage());
            }
        %>

        <table>
            <thead>
                <tr>
                    <th>Customer</th>
                    <th>Cust. Type</th>
                    <th>Purchase</th>
                    <th>Discount</th>
                </tr>
            </thead>
            <tbody>
                <% for (int i = 0; i < csvData.size(); i++) { %>
                    <tr>
                        <%
                            String[] rowData = csvData.get(i);
                            String customer = rowData[0];
                            String custType = rowData[1];
                            int purchase = Integer.parseInt(rowData[2]);
                            double discount = 0.0;
                            if (custType.equalsIgnoreCase("Cash")) {
                                discount = purchase * 0.1;
                            }
                        %>
                        <td><%= customer %></td>
                        <td><%= custType %></td>
                        <td><%= purchase %></td>
                        <td><%= String.format("%.2f", discount) %></td>
                    </tr>
                <% } %>
            </tbody>
        </table>

    </body>
</html>
```



	A	B	C
1	10001	Credit	5000
2	10002	Cash	2500
3	10003	Credit	3000
4	10004	Cash	3200
5	10005	Credit	1200
6	10006	Cash	6000
7	10007	Credit	7400
8	10008	Cash	800

Output:



Read CSV files and populate it into HTML's table

Customer	Cust. Type	Purchase	Discount
10001	Credit	5000	0.00
10002	Cash	2500	250.00
10003	Credit	3000	0.00
10004	Cash	3200	320.00
10005	Credit	1200	0.00
10006	Cash	6000	600.00
10007	Credit	7400	0.00
10008	Cash	800	80.00