1 Exceptions and exception handling

In this chapter we are going to discuss exceptions and their handling. We will start the discussion by explaining exceptions and exception handling in general terms. Subsequent to that we will talk about how to work with exceptions and their handling in a web development environment.

1.1 What is an exception?

An exception is an unexpected condition that can take place during programme execution. So this is a condition that a program was not designed to work with. For example you can have a program that adds numbers. If a user provides nonnumeric input instead, the program will malfunction. It is important to design our program in such a way that they are able to continue working even when the unexpected happens.

Exceptions are mainly divided into two, we have runtime exceptions and compile time exceptions. Runtime exceptions are exceptions that can only be detected during runtime, the time when a program is running. They are logical in nature and the programmer is expected to know how to handle them prior execution. Going back to the addition problem, the programmer needs to think ahead as to what must take place when a user enters a nonnumeric value for a number. This is called exception handling. Runtime exceptions are also known as unchecked exceptions. This means they are not checked at program compile time. So a programmer is not forced to handle them.

Compile time exceptions are exceptions that the compiler detects at compile time if they are not handled. The compiler forces the programmer to handle them before proceeding any further. These kinds of exceptions are also known as checked exceptions. Examples of compile time exceptions are all your file I/O (Input/Output) operations.

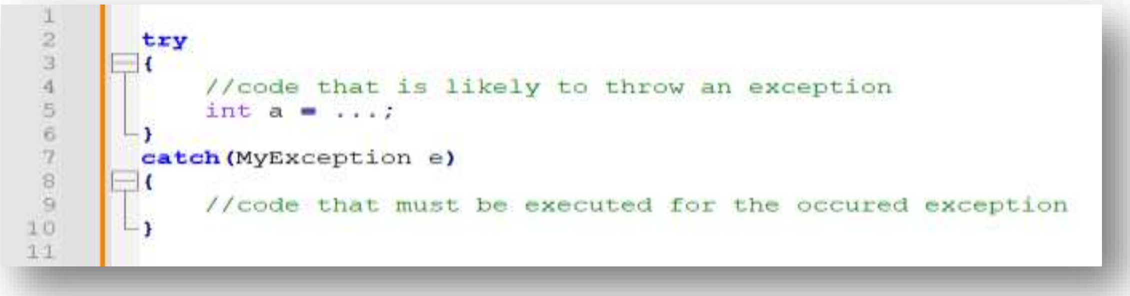
1.2 What is exception handling?

Exception handling is the mechanism used to handle exceptions when they occur. Through exception handling we avoid a situation whereby the behavior of a program becomes unpredictable due to the occurrence of an exception. So we manage the exception and bring the program back to normalcy. There are two ways of handling exceptions, namely:

- Usage of a try catch block; or
- Design a method that rethrows an exception.

1.2.1 Try catch block

The try catch block is used to handle an exception. Code that is likely to throw an exception is kept inside the try block, and code that must handle the exception is put inside the catch block. Below is the generic syntax of a try catch block.

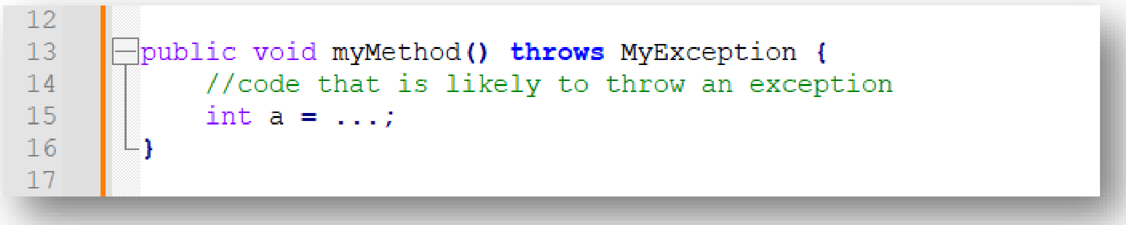


```

try
{
    //code that is likely to throw an exception
    int a = ...;
}
catch(MyException e)
{
    //code that must be executed for the occured exception
}

```

When an exception takes place, an exception object is created. The object is then thrown into the catch blocks. We can have many catch blocks. Each catch block is responsible for handling one exception type. Then a catch block matching the thrown exception is sought. If one is found, the code inside that block is executed. If none is found, the program crashes or simply behaves in an unpredictable manner.

1.2.2 Exception rethrowing methods

Another way of handling an exception is through an exception rethrow. This is accomplished throw a method that executes a piece of code that is likely to throw an exception. Rather than handling the exception inside the method, the method gets to rethrow the exception. This approach defers exception handling to other users of the method. The figure below shows exception rethrow through a method.



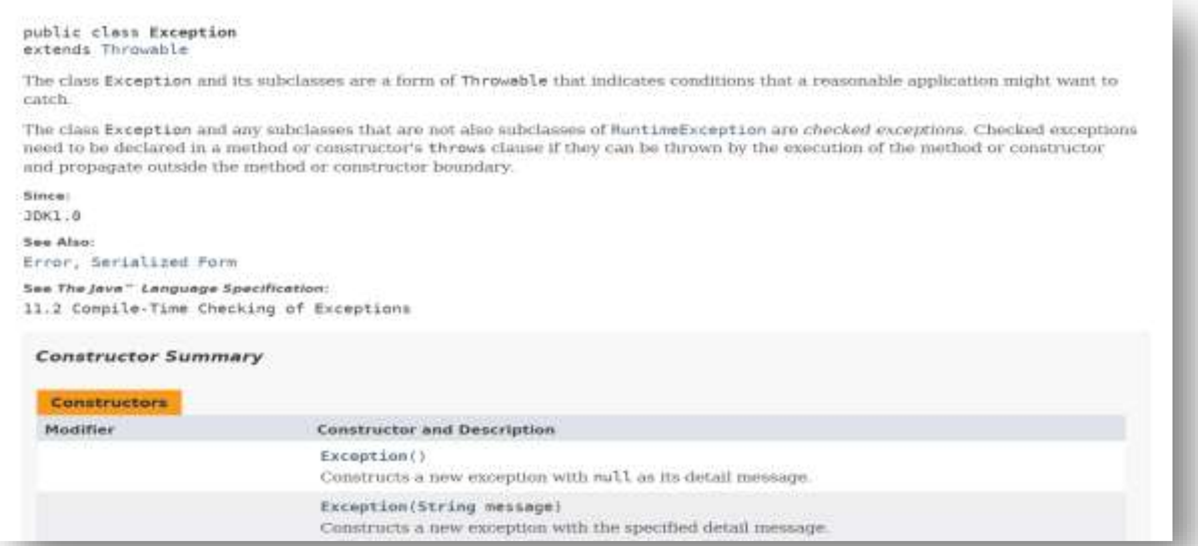
```java
public void myMethod() throws MyException {
    //code that is likely to throw an exception
    int a = ...;
}
```

1.3 Predefined exceptions

The Java programming language comes along with predefined classes, some of which some represent exceptions. These are the exceptions that the JDK creators encountered when developing the language. Some of the exception classes contained in the JDK is the Exception, RuntimeException, and NumberFormatException class.

1.3.1 Exception class

Below is a brief description of the class which is located in the java.lang package



```
public class Exception
extends Throwable
```

The class Exception and its subclasses are a form of Throwable that indicates conditions that a reasonable application might want to catch.

The class Exception and any subclasses that are not also subclasses of RuntimeException are *checked exceptions*. Checked exceptions need to be declared in a method or constructor's throws clause if they can be thrown by the execution of the method or constructor and propagate outside the method or constructor boundary.

Since:
JDK1.0

See Also:
Error, Serialized Form

See *The Java™ Language Specification*:
11.2 Compile-Time Checking of Exceptions

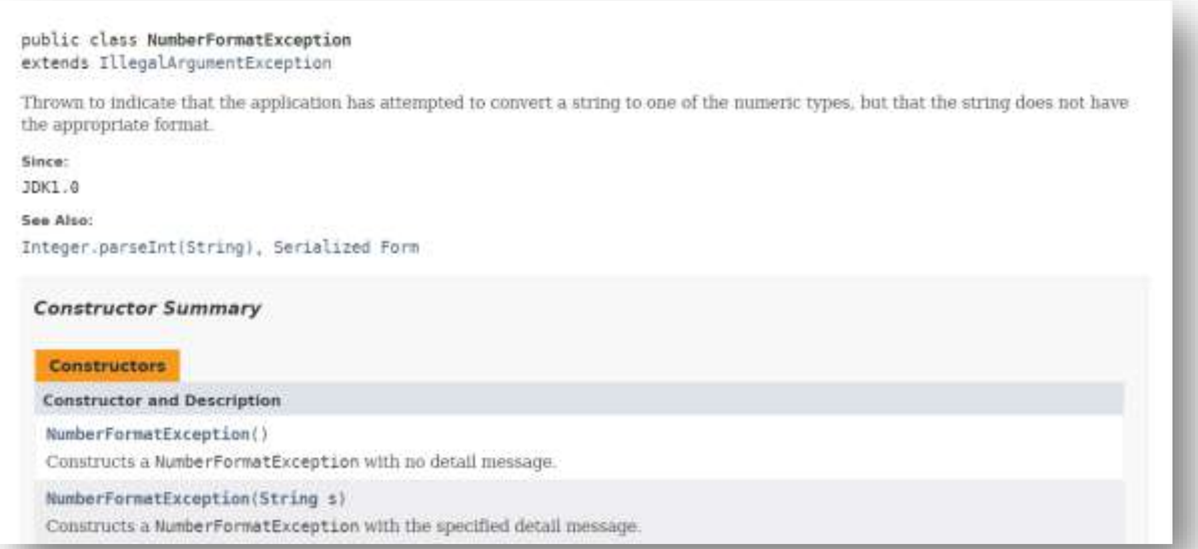
Constructor Summary

Constructors

Modifier	Constructor and Description
	Exception() Constructs a new exception with null as its detail message.
	Exception(String message) Constructs a new exception with the specified detail message.

1.3.2 NumberFormatException class

Below is a brief description of the class which is located in the java.lang package



```
public class NumberFormatException
extends IllegalArgumentException
```

Thrown to indicate that the application has attempted to convert a string to one of the numeric types, but that the string does not have the appropriate format.

Since:
JDK1.0

See Also:
Integer.parseInt(String), Serialized Form

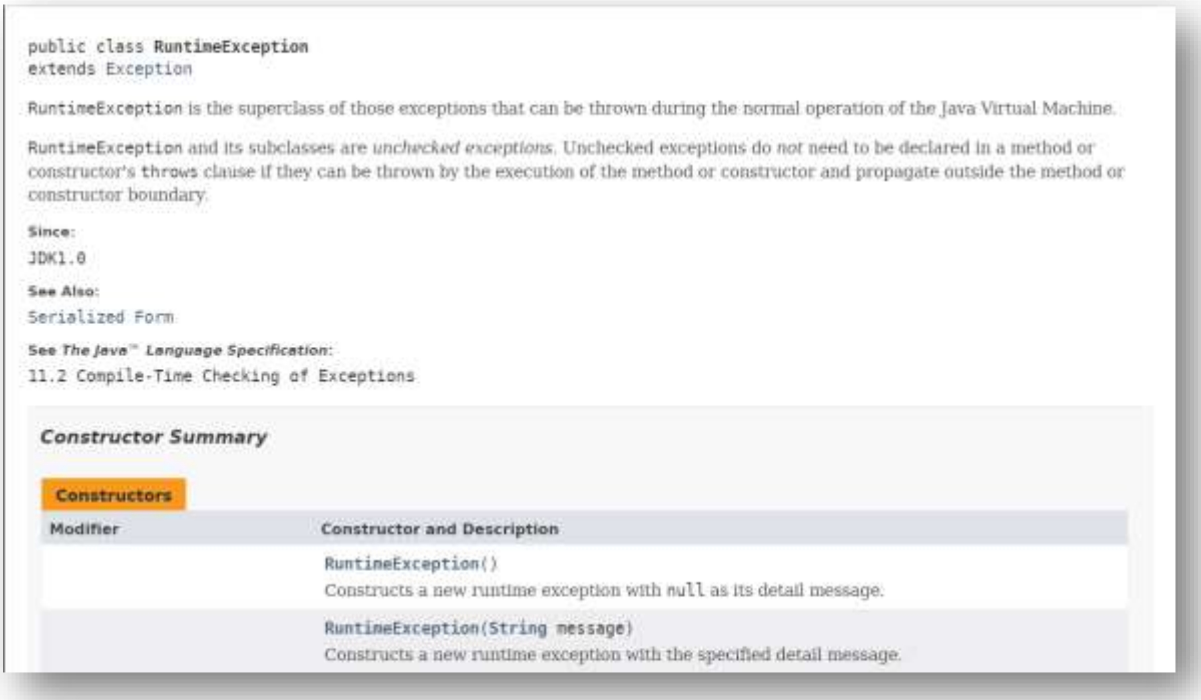
Constructor Summary

Constructors

Constructor and Description
NumberFormatException() Constructs a NumberFormatException with no detail message.
NumberFormatException(String s) Constructs a NumberFormatException with the specified detail message.

1.3.3 RuntimeException

Below is a brief description of the class which is located in the java.lang package



```
public class RuntimeException
extends Exception
```

`RuntimeException` is the superclass of those exceptions that can be thrown during the normal operation of the Java Virtual Machine.

`RuntimeException` and its subclasses are *unchecked exceptions*. Unchecked exceptions do *not* need to be declared in a method or constructor's `throws` clause if they can be thrown by the execution of the method or constructor and propagate outside the method or constructor boundary.

Since:
JDK1.0

See Also:
Serialized Form

See *The Java™ Language Specification*:
11.2 Compile-Time Checking of Exceptions

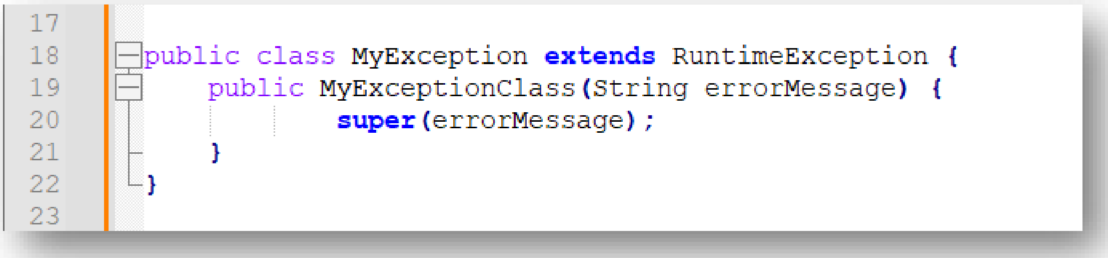
Constructor Summary

Constructors

Modifier	Constructor and Description
	`RuntimeException()` Constructs a new runtime exception with `null` as its detail message.
	`RuntimeException(String message)` Constructs a new runtime exception with the specified detail message.

1.4 User-defined exceptions

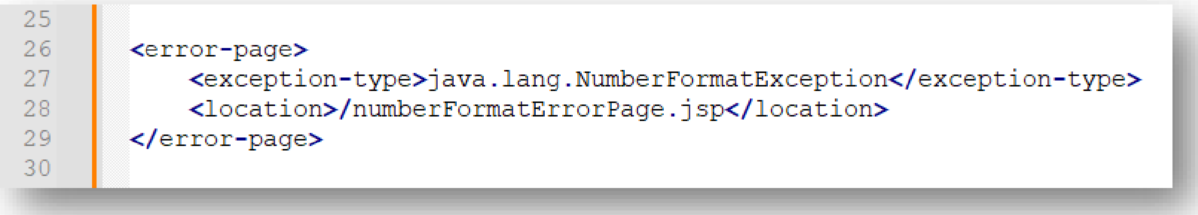
As a programmer you can also define your own exception classes. These classes will simply be there to represent a specific condition that is not catered for by the predefined exception classes. The handling of the exception will work the same as that of predefined exceptions. The generic syntax for defining a user-defined exception is as follows:



```java
public class MyException extends RuntimeException {
    public MyExceptionClass(String errorMessage) {
        super(errorMessage);
    }
}
```

1.5 How to handle exceptions in a web development environment?

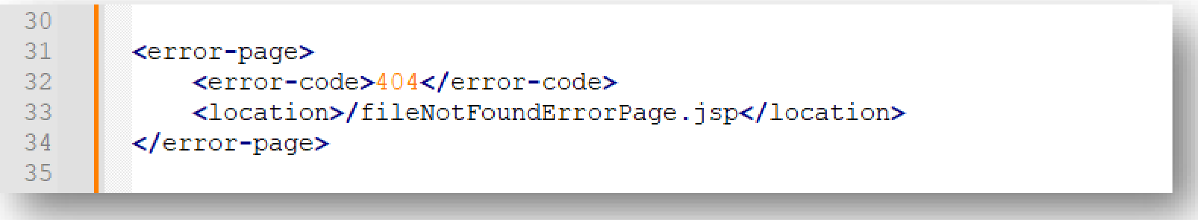
In the web development environment we use a combination of the DD and corresponding files to handle exceptions. In the DD we state the exceptions we want to handle, and also the files that must be called to handle those exceptions. In the DD, an exception can be referenced either by its fully qualified name or code. For example, if we want to catch the NumberFormatException we will have the following in the DD:



```
<error-page>
    <exception-type>java.lang.NumberFormatException</exception-type>
    <location>/numberFormatErrorPage.jsp</location>
</error-page>
```

The **<error-page>** element is used to represent exception handling. Inside the **<error-page>** element we define the type of error we want to handle, **<exception-type>**, and the file that will handle the error, **<location>**.

The exception code is used if we know the code that represents an error. For example, if we were to work with the famous 404 FileNotFoundException, we could represent it as follows in the DD:



```
<error-page>
    <error-code>404</error-code>
    <location>/fileNotFoundErrorPage.jsp</location>
</error-page>
```

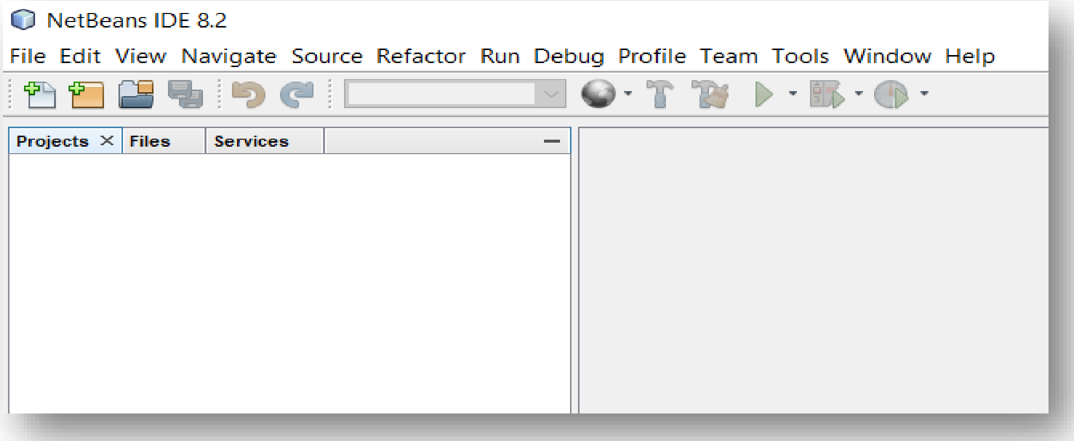
Again we define the error handling file inside and the **location** element.

1.6 Example

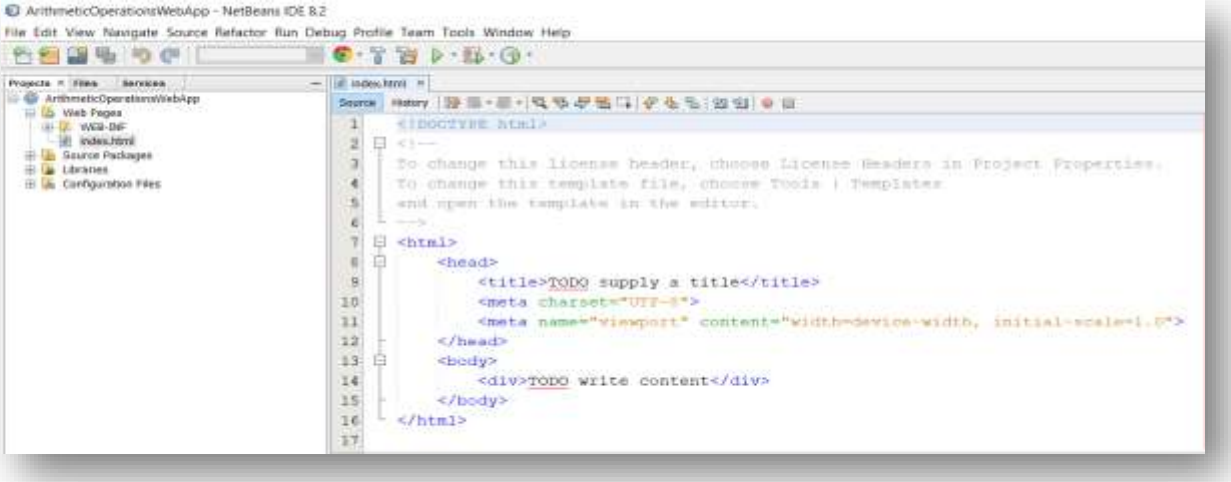
Create a robust web application that will allow a user to perform a preferred arithmetic operation on two numbers. If a user enters an invalid input, the application must be able to handle the situation.

Solution

Launch NetBeans

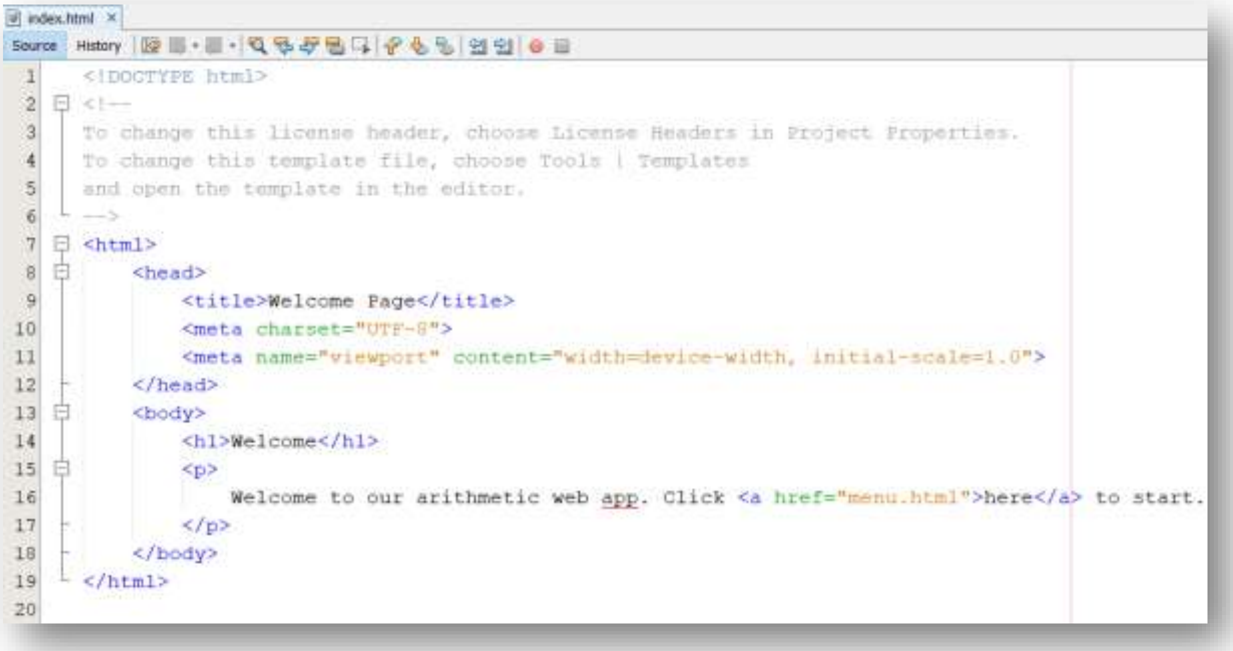

NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects
Files
Services

Create a web application project called **ArithmeticOperationsWebApp**


ArithmeticOperationsWebApp - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects
Files
Services
ArithmeticOperationsWebApp
Web Pages
WEB-INF
index.html
Source Packages
Libraries
Configuration Files
index.html
Source
History

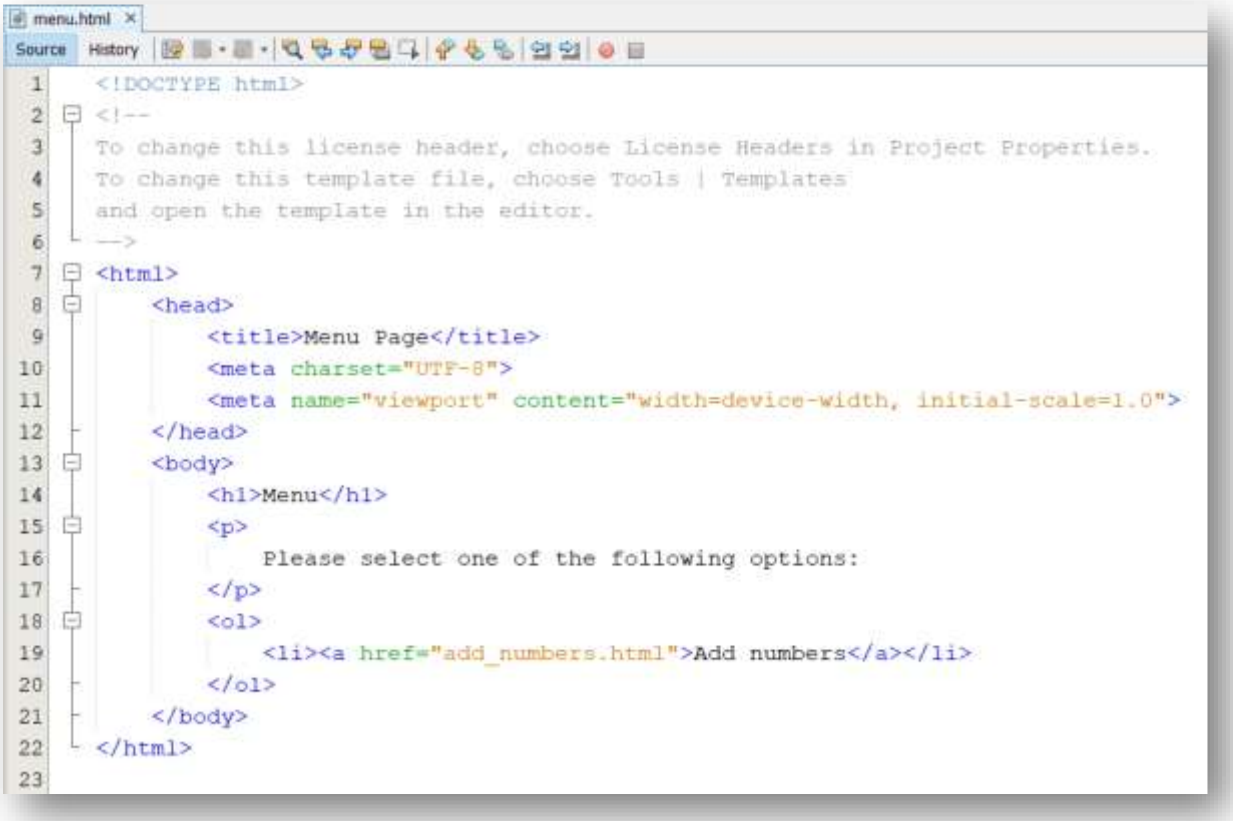
```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
    <head>
        <title>TODO supply a title</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <div>TODO write content</div>
    </body>
</html>
```

Edit the index page



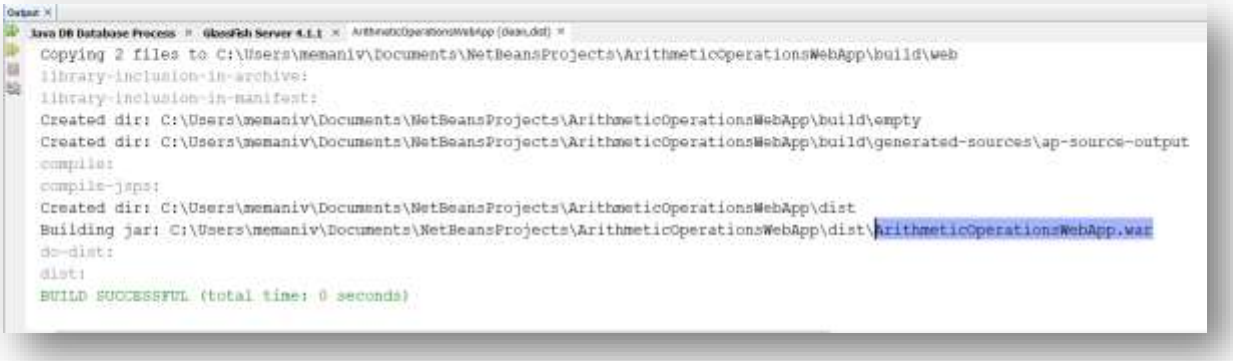
```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
    <head>
        <title>Welcome Page</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <h1>Welcome</h1>
        <p>
            Welcome to our arithmetic web app. Click <a href="menu.html">here</a> to start.
        </p>
    </body>
</html>
```

Create the **menu.html** page



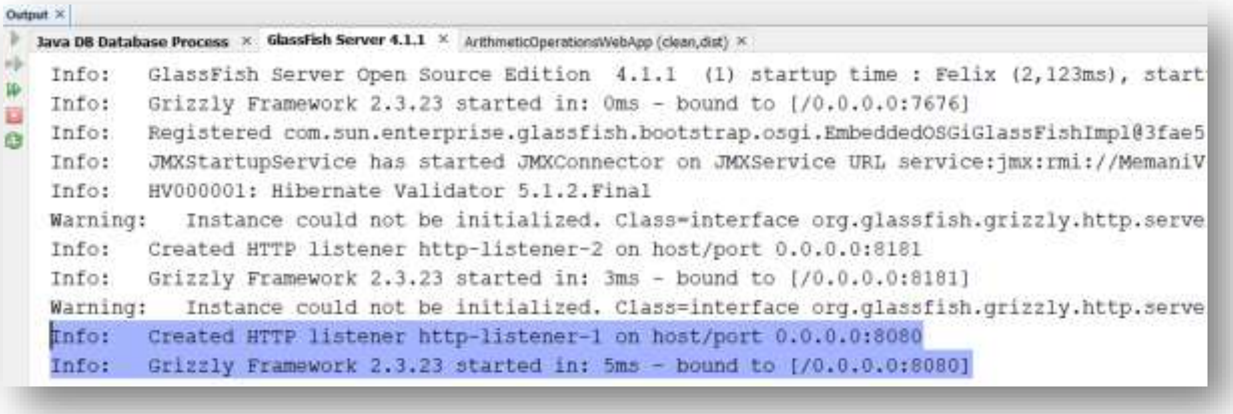
```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
    <head>
        <title>Menu Page</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <h1>Menu</h1>
        <p>
            Please select one of the following options:
        </p>
        <ol>
            <li><a href="add_numbers.html">Add numbers</a></li>
        </ol>
    </body>
</html>
```

Compile the project


Output ×
Java DB Database Process ×
GlassFish Server 4.1.1 ×
ArithmeticOperationsWebApp (clean,dist) ×

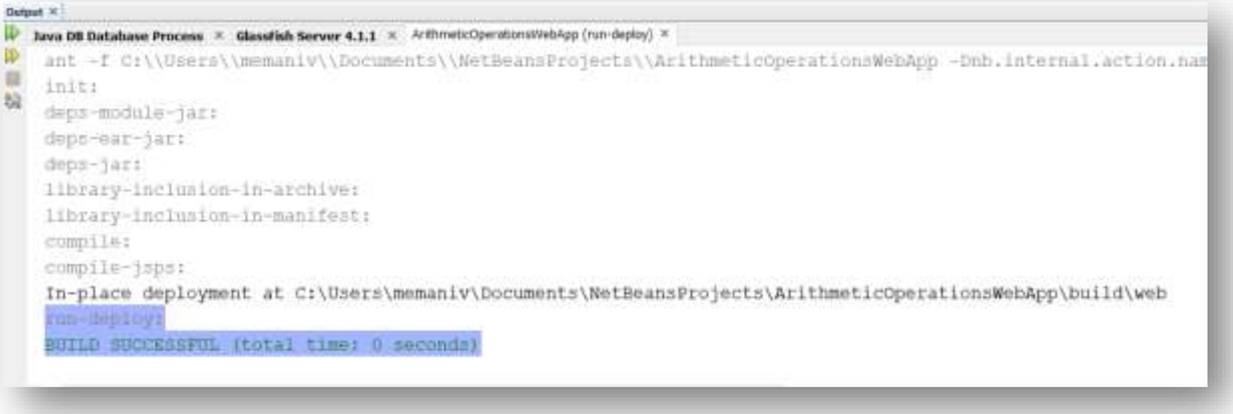
```
Copying 2 files to C:\Users\memaniv\Documents\NetBeansProjects\ArithmeticOperationsWebApp\build\web
library-inclusion-in-archive:
library-inclusion-in-manifest:
Created dir: C:\Users\memaniv\Documents\NetBeansProjects\ArithmeticOperationsWebApp\build\empty
Created dir: C:\Users\memaniv\Documents\NetBeansProjects\ArithmeticOperationsWebApp\build\generated-sources\ap-source-output
compile:
compile-jsps:
Created dir: C:\Users\memaniv\Documents\NetBeansProjects\ArithmeticOperationsWebApp\dist
Building jar: C:\Users\memaniv\Documents\NetBeansProjects\ArithmeticOperationsWebApp\dist\ArithmeticOperationsWebApp.war
do-dist:
dist:
BUILD SUCCESSFUL (total time: 0 seconds)
```

Start GlassFish


Output ×
Java DB Database Process ×
GlassFish Server 4.1.1 ×
ArithmeticOperationsWebApp (clean,dist) ×

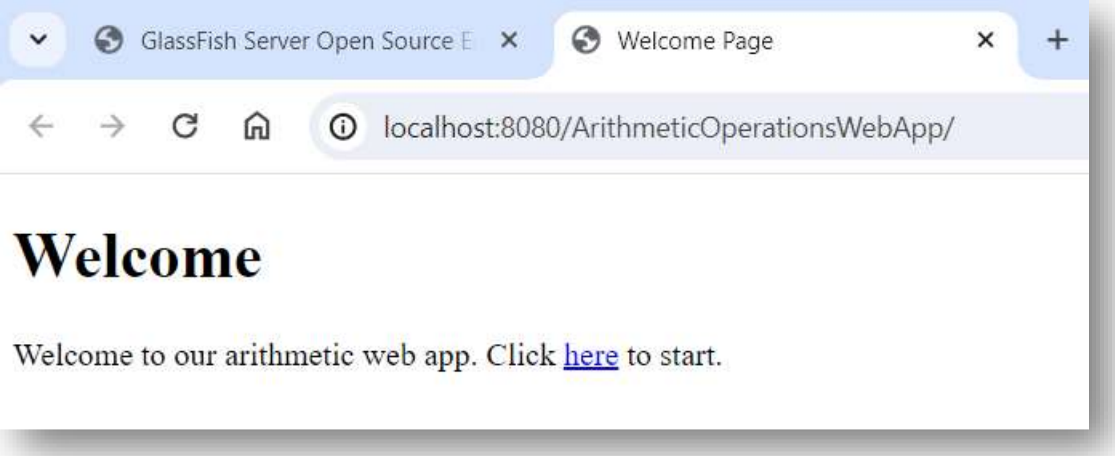
```
Info:   GlassFish Server Open Source Edition  4.1.1  (1) startup time : Felix (2,123ms), start
Info:   Grizzly Framework 2.3.23 started in: 0ms - bound to [/0.0.0.0:7676]
Info:   Registered com.sun.enterprise.glassfish.bootstrap.osgi.EmbeddedOSGiGlassFishImpl@3fae5
Info:   JMXStartupService has started JMXConnector on JMXService URL service:jmx:rmi://MemaniV
Info:   HV000001: Hibernate Validator 5.1.2.Final
Warning:   Instance could not be initialized. Class=interface org.glassfish.grizzly.http.serve
Info:   Created HTTP listener http-listener-2 on host/port 0.0.0.0:8181
Info:   Grizzly Framework 2.3.23 started in: 3ms - bound to [/0.0.0.0:8181]
Warning:   Instance could not be initialized. Class=interface org.glassfish.grizzly.http.serve
Info:   Created HTTP listener http-listener-1 on host/port 0.0.0.0:8080
Info:   Grizzly Framework 2.3.23 started in: 5ms - bound to [/0.0.0.0:8080]
```

Deploy the application


Output ×
Java DB Database Process ×
GlassFish Server 4.1.1 ×
ArithmeticOperationsWebApp (run-deploy) ×

```
ant -f C:\\Users\\memaniv\\Documents\\NetBeansProjects\\ArithmeticOperationsWebApp -Dnb.internal.action.nam
init:
deps-module-jar:
deps-ear-jar:
deps-jar:
library-inclusion-in-archive:
library-inclusion-in-manifest:
compile:
compile-jsps:
In-place deployment at C:\Users\memaniv\Documents\NetBeansProjects\ArithmeticOperationsWebApp\build\web
run-deploy:
BUILD SUCCESSFUL (total time: 0 seconds)
```

Run the application

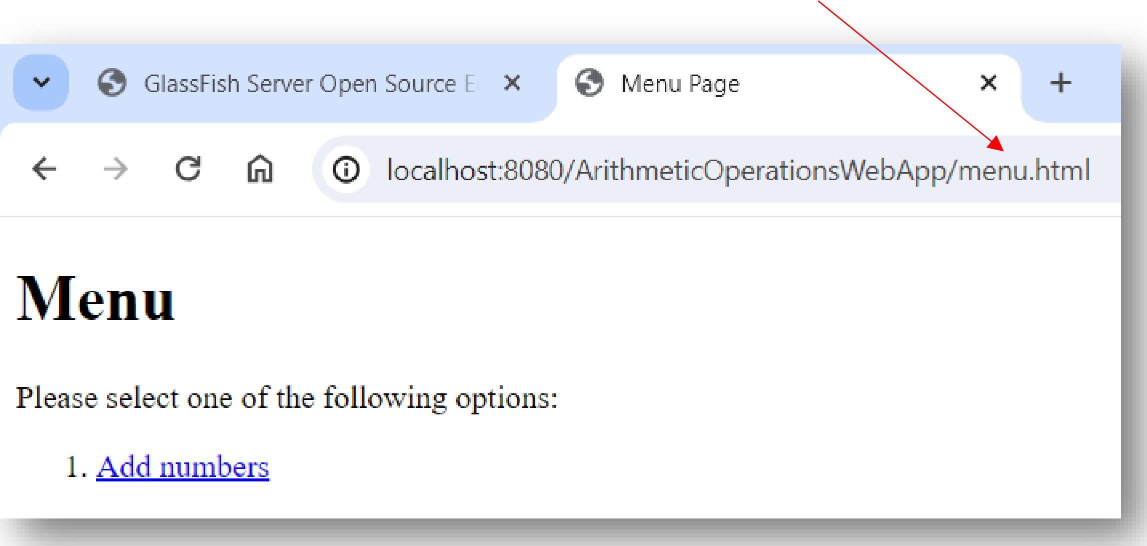

GlassFish Server Open Source E
Welcome Page
localhost:8080/ArithmeticOperationsWebApp/

Welcome

Welcome to our arithmetic web app. Click here to start.

Click on the link

The **menu.html** page is invoked by the **index.html**

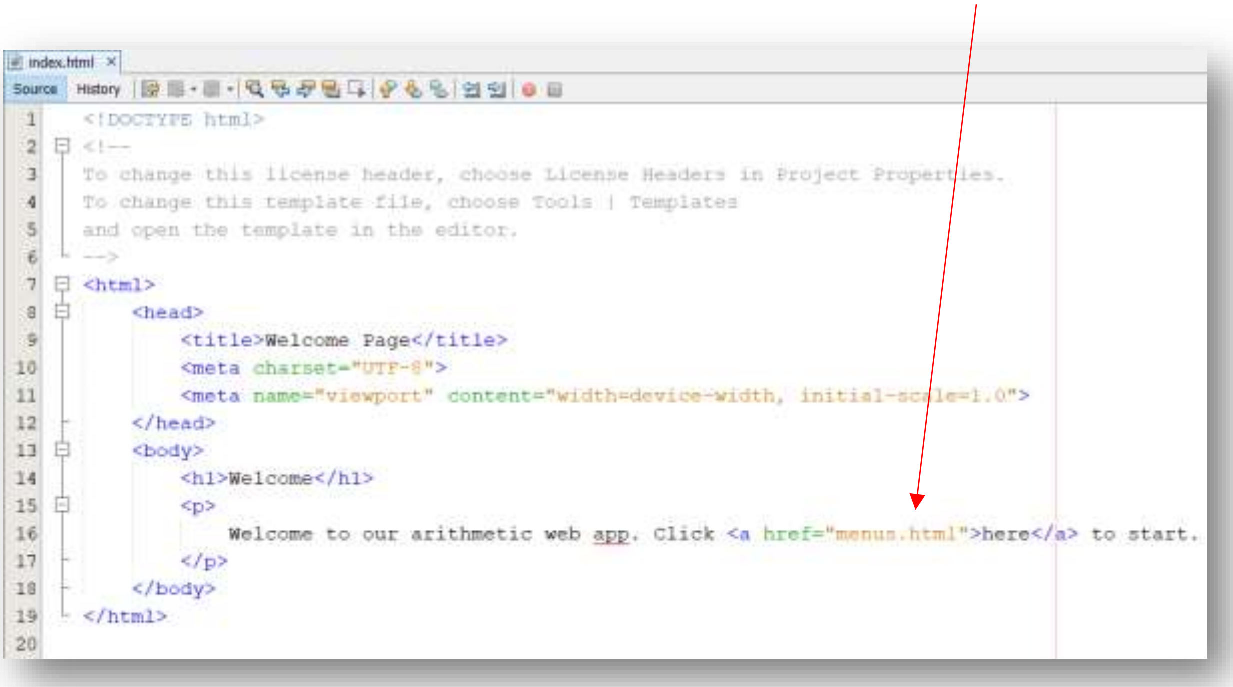

GlassFish Server Open Source E
Menu Page
localhost:8080/ArithmeticOperationsWebApp/menu.html

Menu

Please select one of the following options:

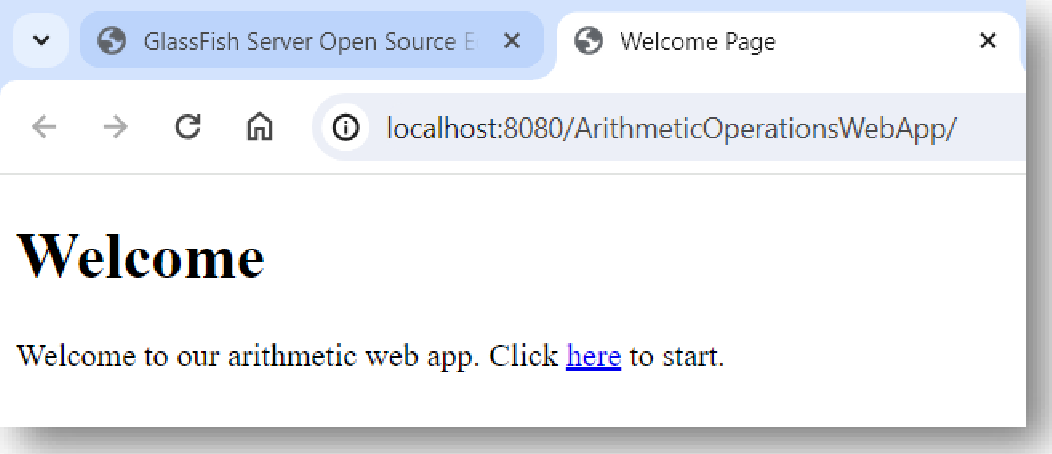
1. Add numbers

Let’s deliberately link the **index.html** to a page that is non-existent, **menus.html**



```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
    <head>
        <title>Welcome Page</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <h1>Welcome</h1>
        <p>
            Welcome to our arithmetic web app. Click <a href="menus.html">here</a> to start.
        </p>
    </body>
</html>
```

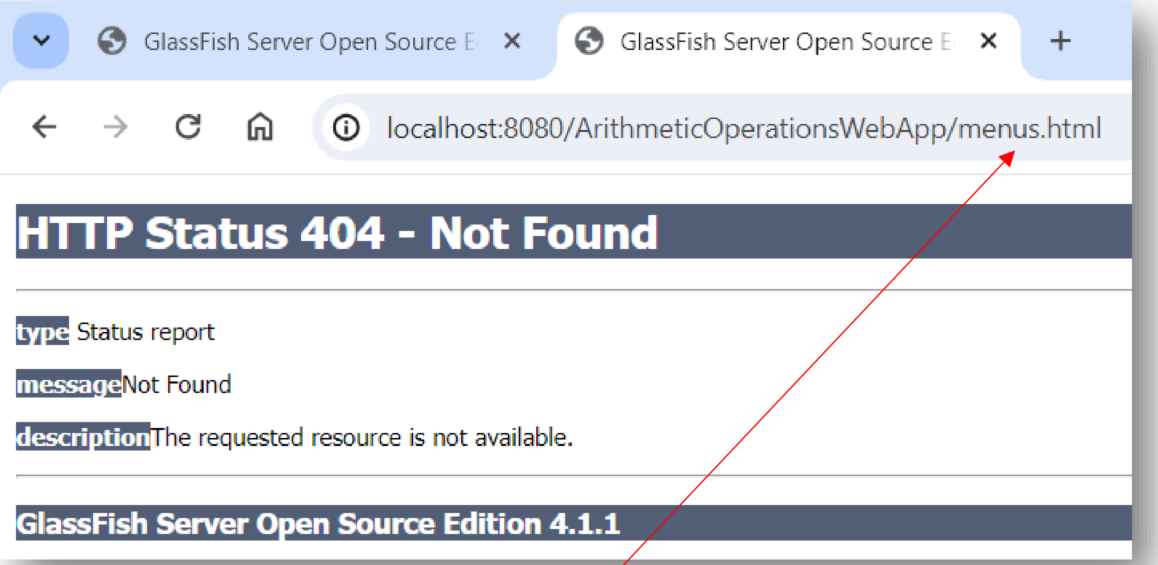
Compile, deploy and run the application


GlassFish Server Open Source E
Welcome Page
localhost:8080/ArithmeticOperationsWebApp/

Welcome

Welcome to our arithmetic web app. Click here to start.

Click on the link


GlassFish Server Open Source E
GlassFish Server Open Source E
localhost:8080/ArithmeticOperationsWebApp/menus.html

HTTP Status 404 - Not Found

type Status report

messageNot Found

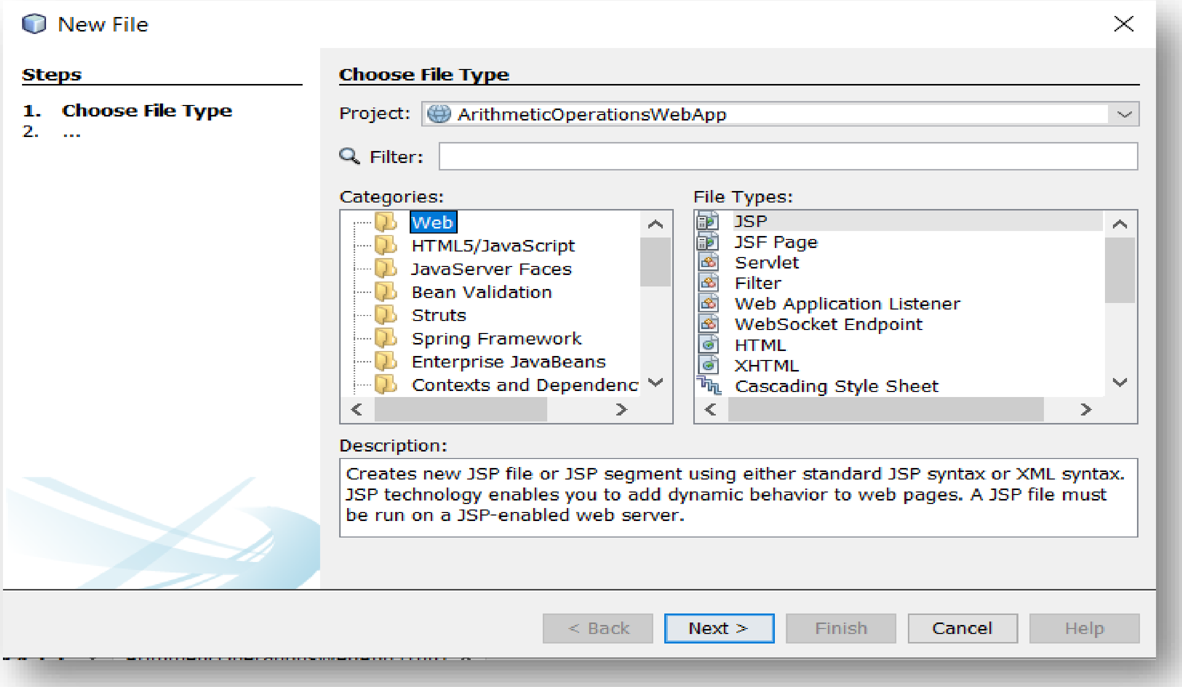
descriptionThe requested resource is not available.

GlassFish Server Open Source Edition 4.1.1

An HTTP exception is thrown. It is the file not found **exception** (**404**). This means the resource that was requested (**menus.html**) was not found. We need to write code in the DD file to handle this exception.

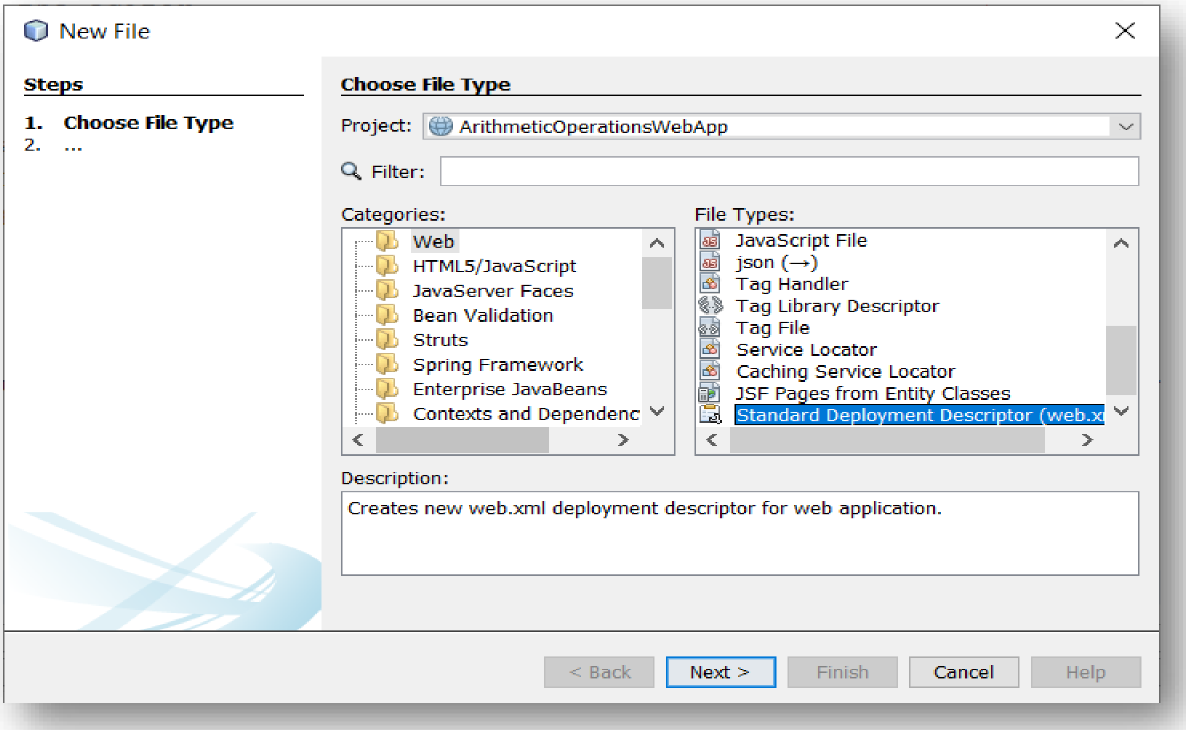
Create a DD file.

- Right-click on the project and select **New | Other**



Creates new JSP file or JSP segment using either standard JSP syntax or XML syntax. JSP technology enables you to add dynamic behavior to web pages. A JSP file must be run on a JSP-enabled web server.

- Select **Web** under **Categories** and **Standard Deployment Descriptor (web.xml)** under **File Types**



New File

Steps

1. **Choose File Type**
2. ...

Choose File Type

Project: ArithmeticOperationsWebApp

Filter:

Categories:

- Web
- HTML5/JavaScript
- JavaServer Faces
- Bean Validation
- Struts
- Spring Framework
- Enterprise JavaBeans
- Contexts and Dependenc

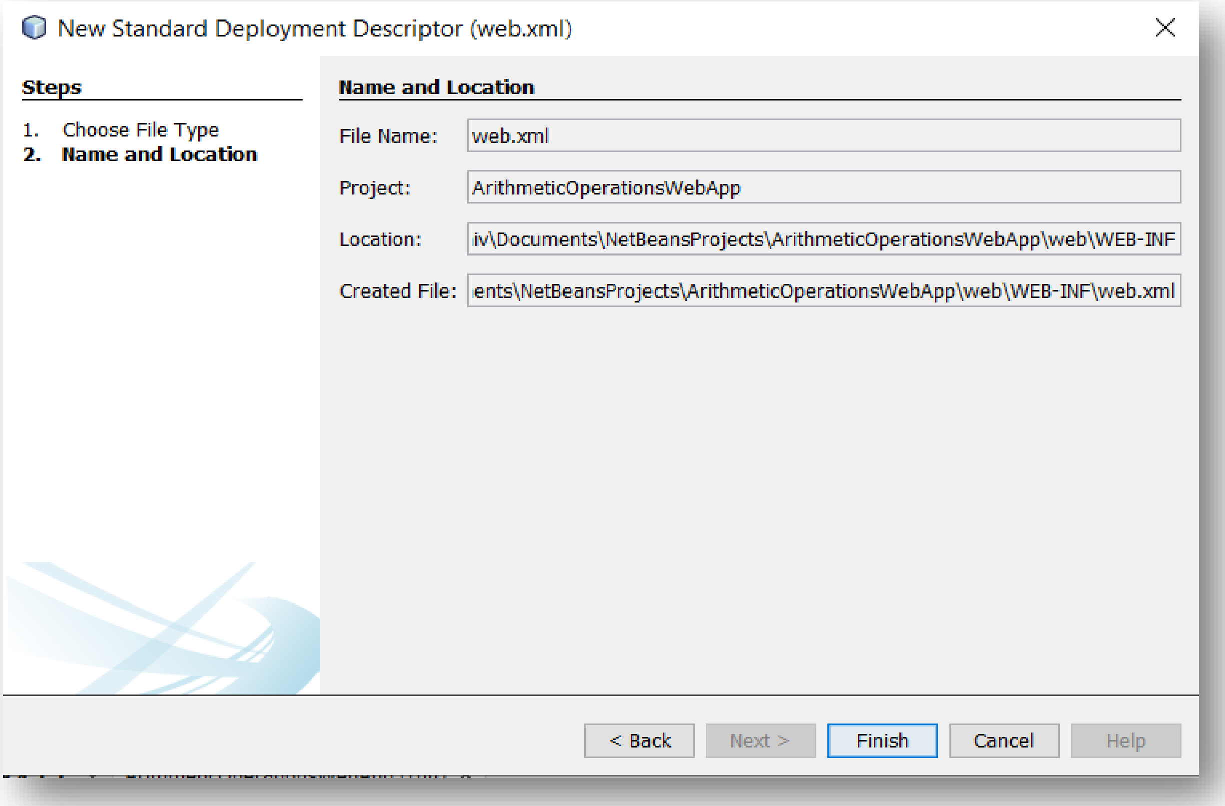
File Types:

- JavaScript File
- json (→)
- Tag Handler
- Tag Library Descriptor
- Tag File
- Service Locator
- Caching Service Locator
- JSF Pages from Entity Classes
- Standard Deployment Descriptor (web.x

Description:

Creates new web.xml deployment descriptor for web application.

- Click Next



New Standard Deployment Descriptor (web.xml)

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

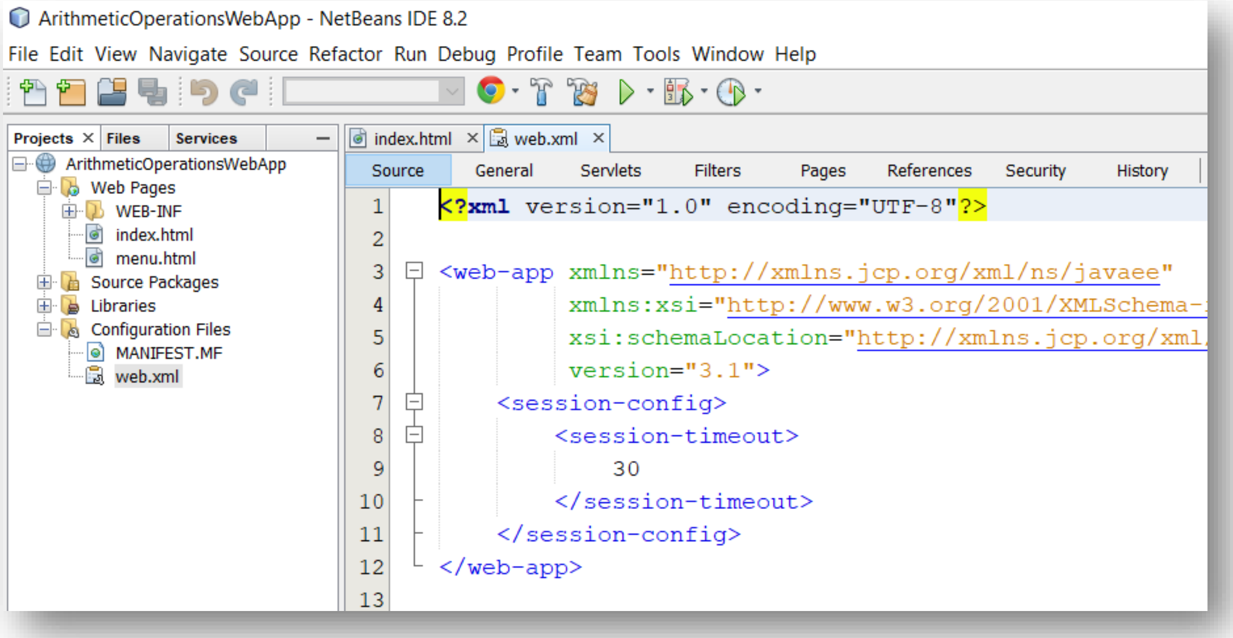
File Name: web.xml

Project: ArithmeticOperationsWebApp

Location: iv\Documents\NetBeansProjects\ArithmeticOperationsWebApp\web\WEB-INF

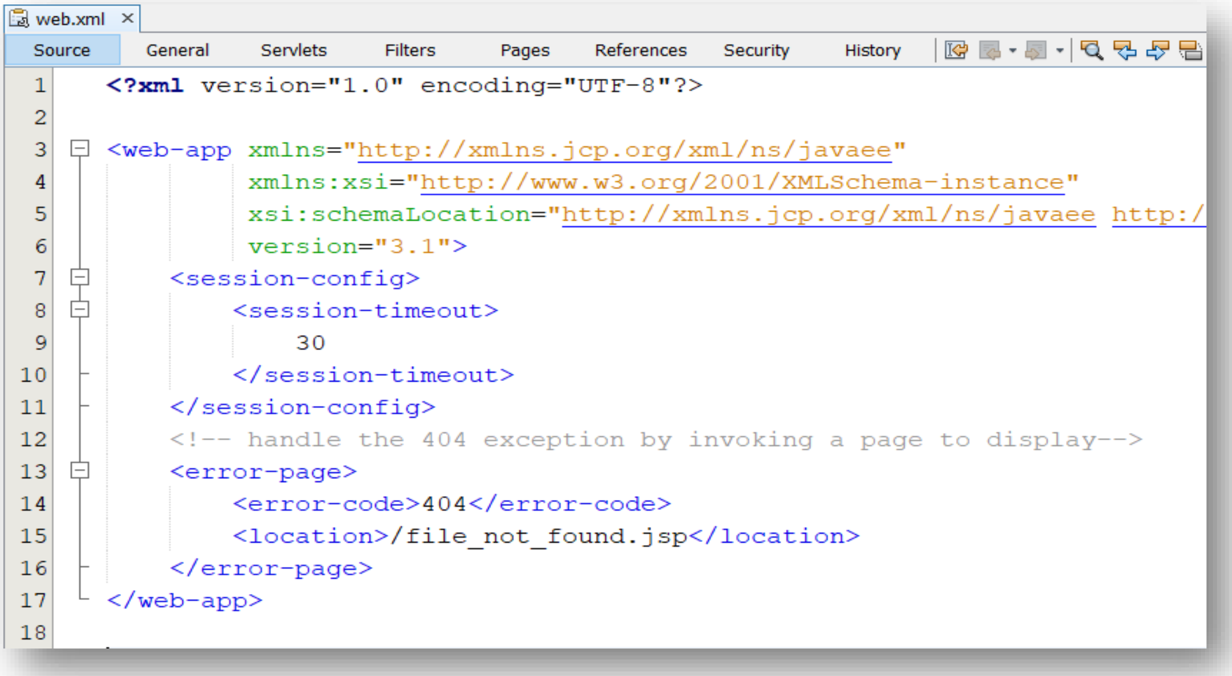
Created File: ents\NetBeansProjects\ArithmeticOperationsWebApp\web\WEB-INF\web.xml

- Click Finish



```xml
<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-i
         xsi:schemaLocation="http://xmlns.jcp.org/xml
         version="3.1">
    <session-config>
        <session-timeout>
            30
        </session-timeout>
    </session-config>
</web-app>

```

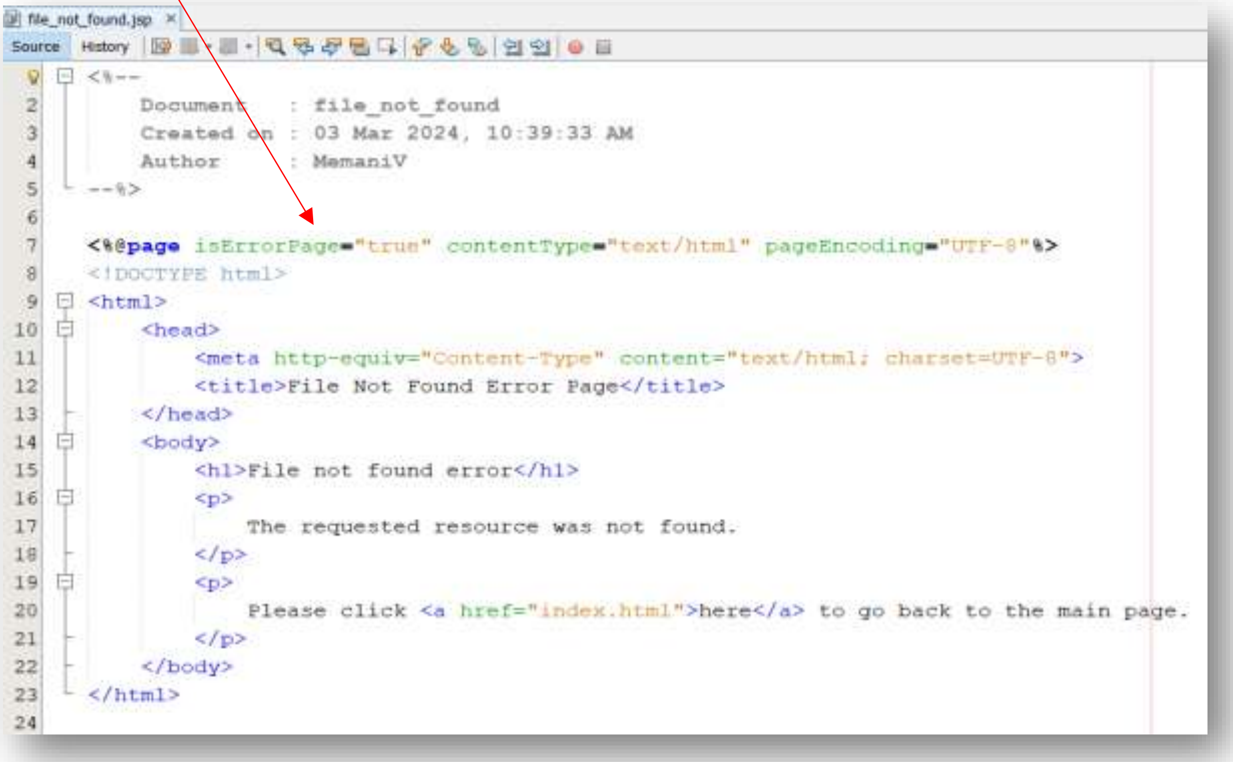
- Handle the **404** error in the DD file



```xml
<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http:/
         version="3.1">
    <session-config>
        <session-timeout>
            30
        </session-timeout>
    </session-config>
    <!-- handle the 404 exception by invoking a page to display-->
    <error-page>
        <error-code>404</error-code>
        <location>/file_not_found.jsp</location>
    </error-page>
</web-app>

```

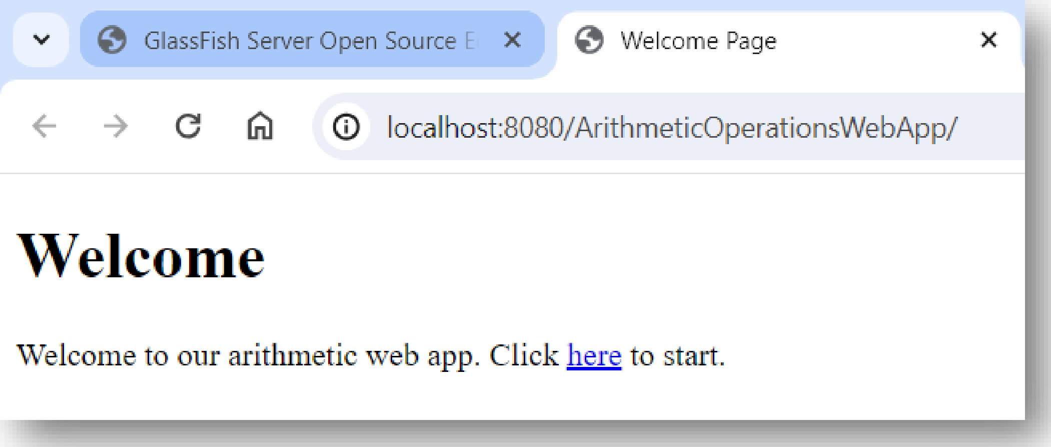
- Create the **file_not_found.jsp** page

This is an error page



```
<%--
    Document   : file_not_found
    Created on : 03 Mar 2024, 10:39:33 AM
    Author     : MemaniV
--%>

<%@page isErrorPage="true" contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>File Not Found Error Page</title>
    </head>
    <body>
        <h1>File not found error</h1>
        <p>
            The requested resource was not found.
        </p>
        <p>
            Please click <a href="index.html">here</a> to go back to the main page.
        </p>
    </body>
</html>
```

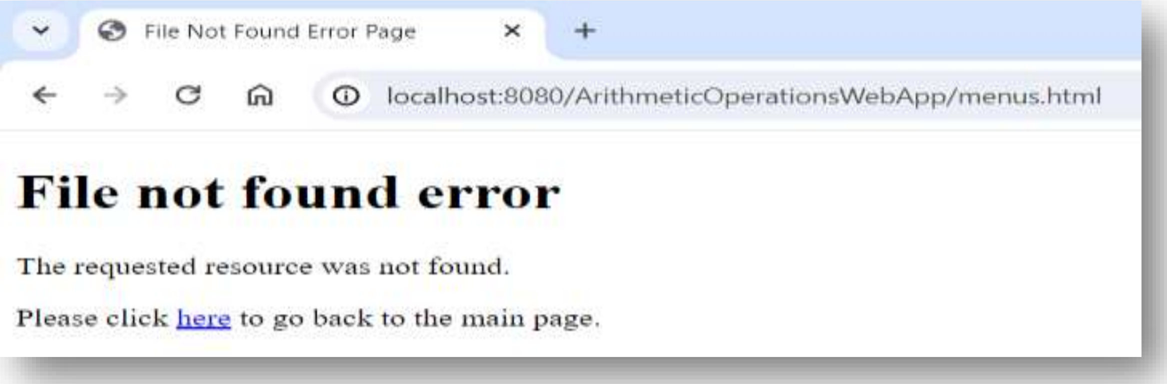
- Clean and Build, Deploy and Run.


GlassFish Server Open Source E
Welcome Page
localhost:8080/ArithmeticOperationsWebApp/

Welcome

Welcome to our arithmetic web app. Click here to start.

- Click on the link

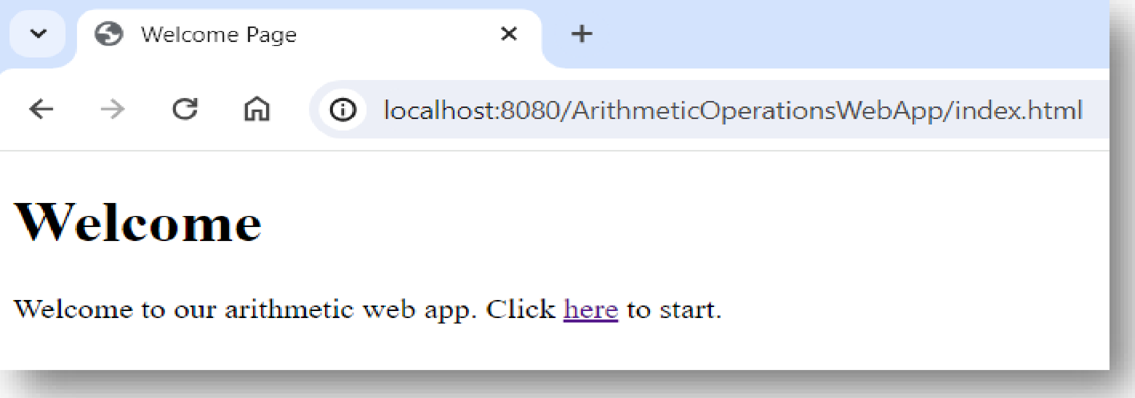

File Not Found Error Page
localhost:8080/ArithmeticOperationsWebApp/menus.html

File not found error

The requested resource was not found.

Please click here to go back to the main page.

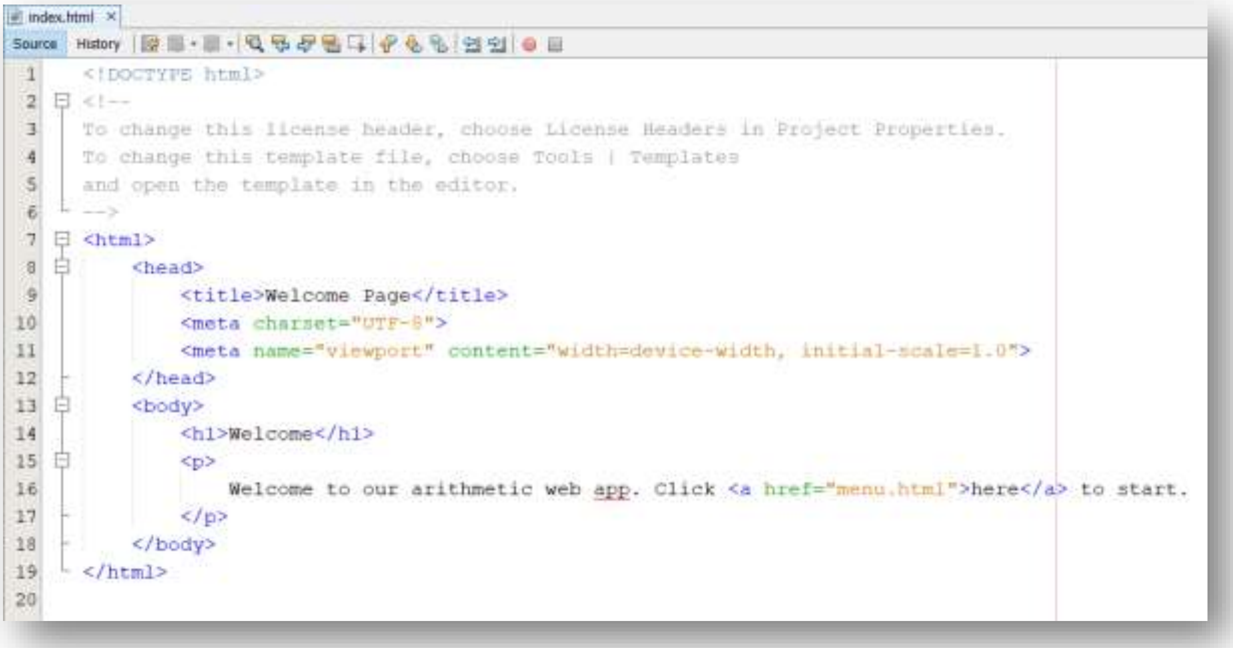
- Click on the link



Welcome

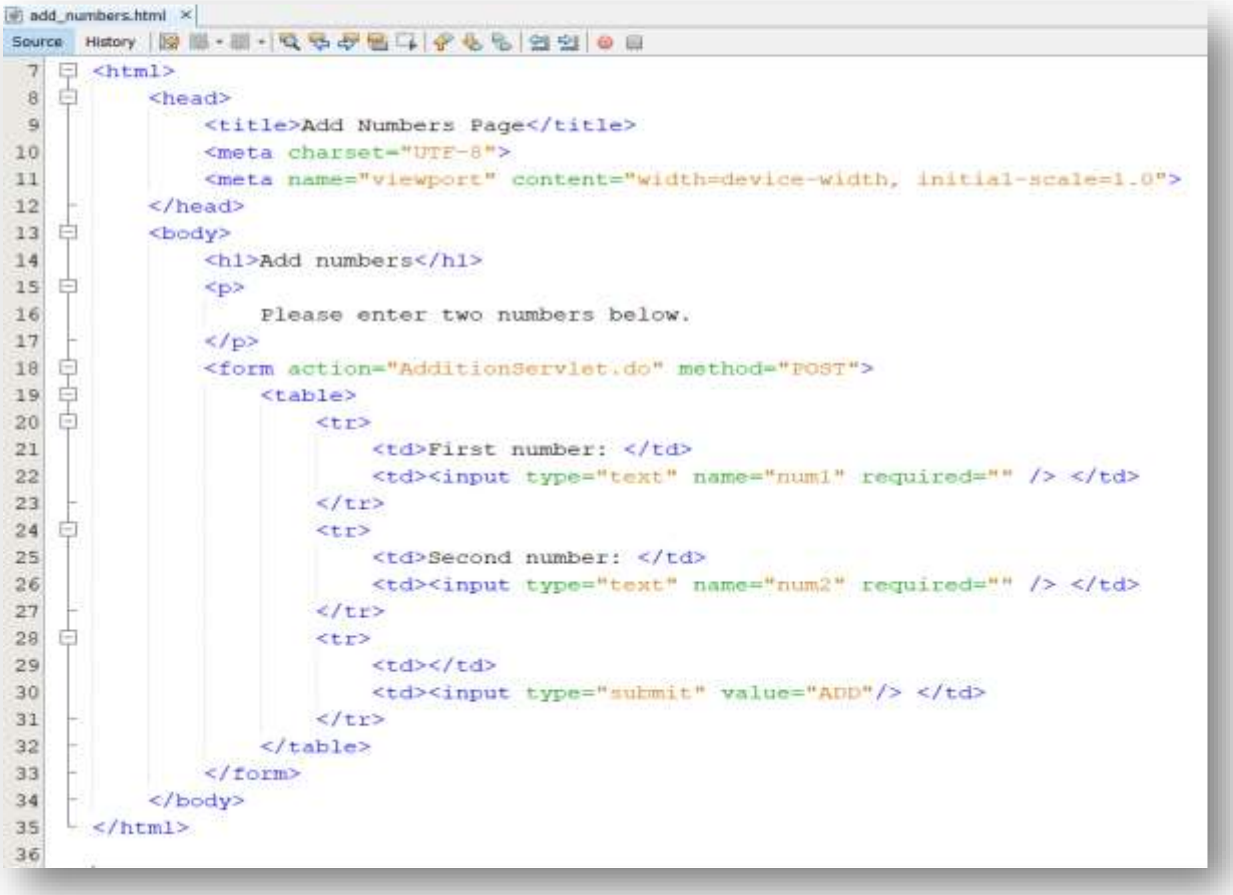
Welcome to our arithmetic web app. Click here to start.

- Change the file name back to **menu.html**.



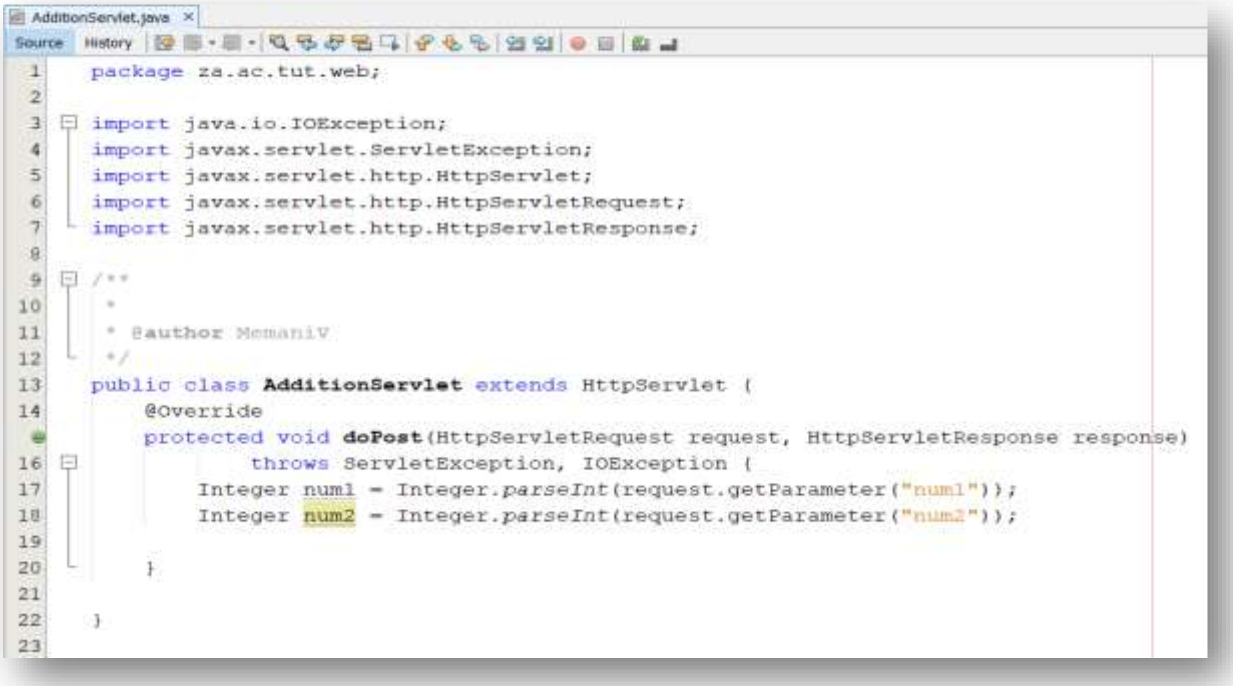
```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
    <head>
        <title>Welcome Page</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <h1>Welcome</h1>
        <p>
            Welcome to our arithmetic web app. Click <a href="menu.html">here</a> to start.
        </p>
    </body>
</html>
```

- Create the **add_numbers.html** page



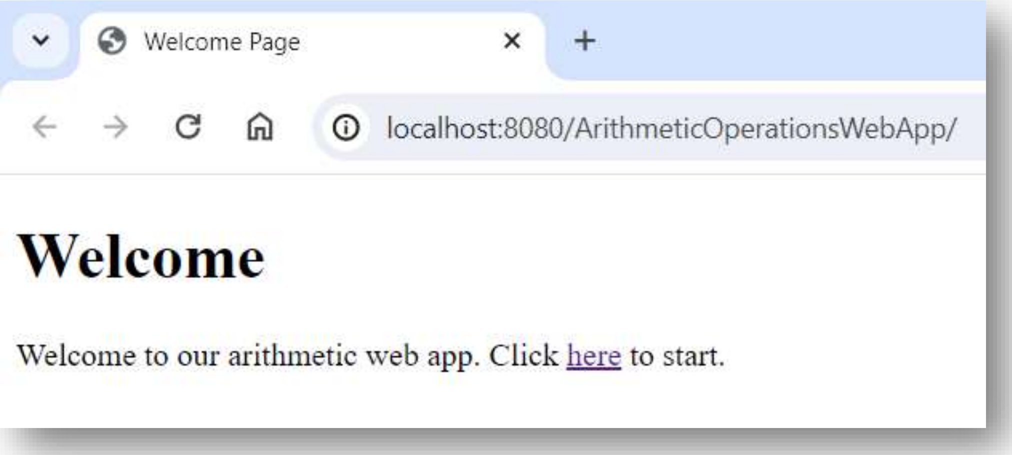
```html
<html>
    <head>
        <title>Add Numbers Page</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <h1>Add numbers</h1>
        <p>
            Please enter two numbers below.
        </p>
        <form action="AdditionServlet.do" method="POST">
            <table>
                <tr>
                    <td>First number: </td>
                    <td><input type="text" name="num1" required="" /> </td>
                </tr>
                <tr>
                    <td>Second number: </td>
                    <td><input type="text" name="num2" required="" /> </td>
                </tr>
                <tr>
                    <td></td>
                    <td><input type="submit" value="ADD"/> </td>
                </tr>
            </table>
        </form>
    </body>
</html>
```

- Create the **AdditionServlet** file



```java
package za.ac.tut.web;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 *
 * @author MemaniV
 */
public class AdditionServlet extends HttpServlet {
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        Integer num1 = Integer.parseInt(request.getParameter("num1"));
        Integer num2 = Integer.parseInt(request.getParameter("num2"));

    }

}
```

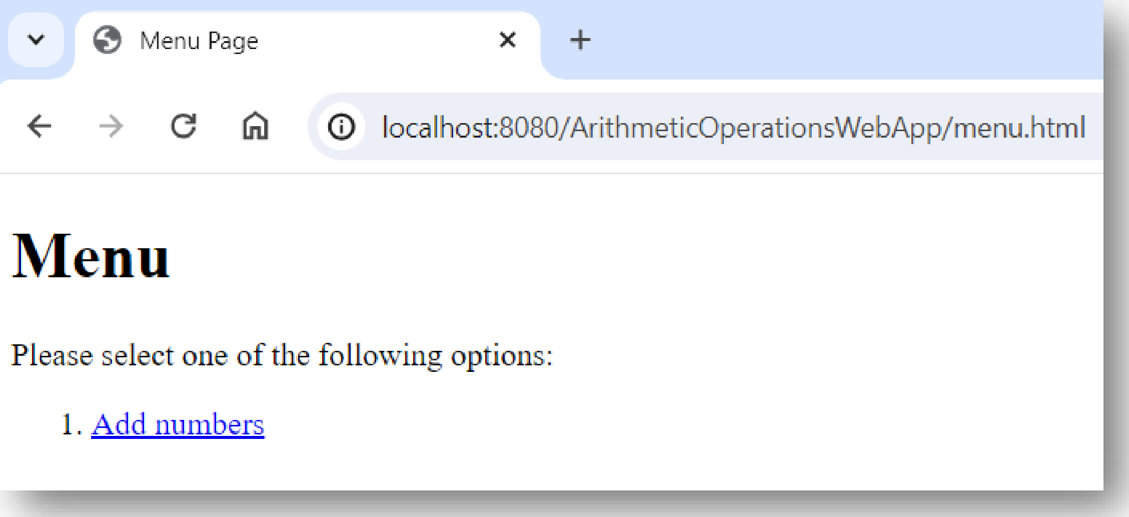
Clean and Build, Deploy and Run



Welcome

Welcome to our arithmetic web app. Click here to start.

Click on the link

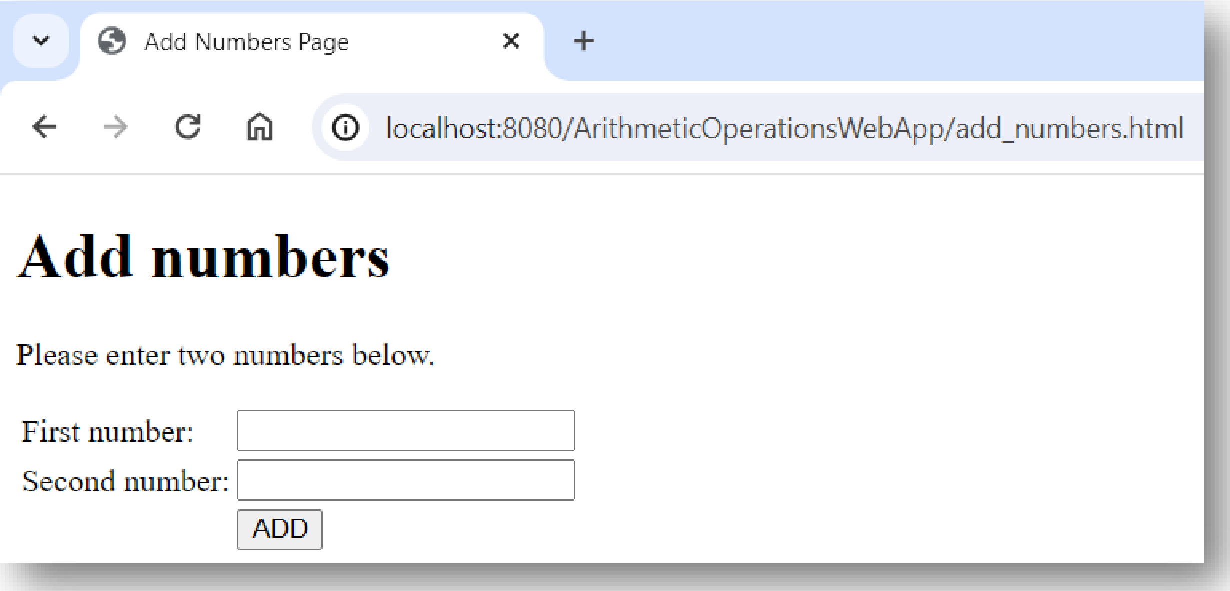


Menu

Please select one of the following options:

1. Add numbers

Click on the link



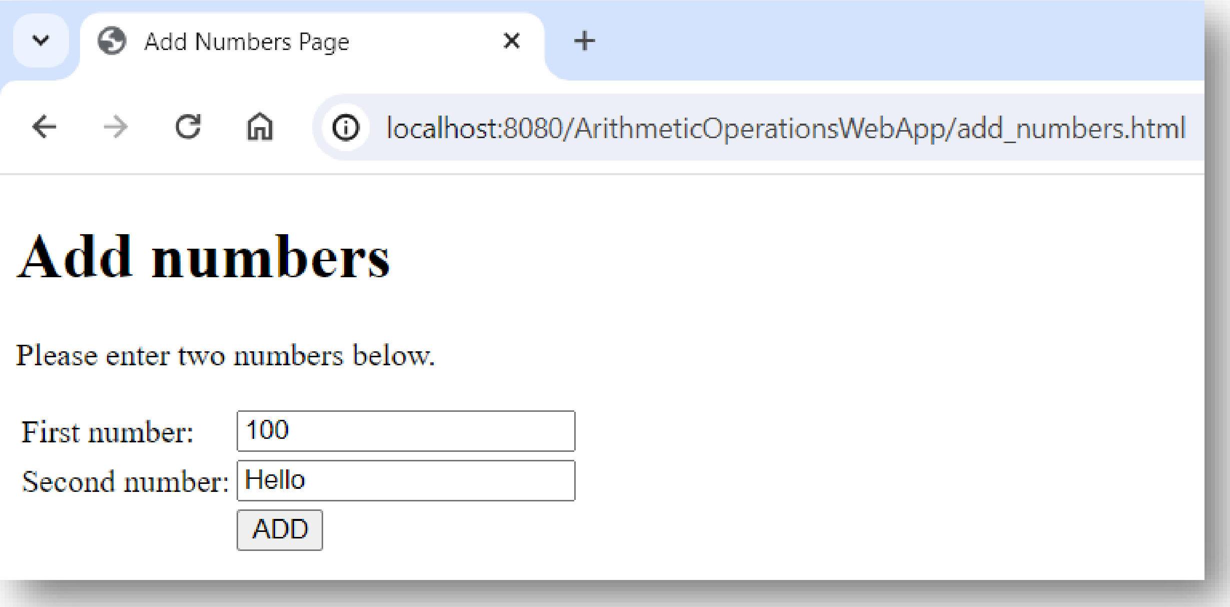
Add numbers

Please enter two numbers below.

First number:

Second number:

Enter a valid number and a string



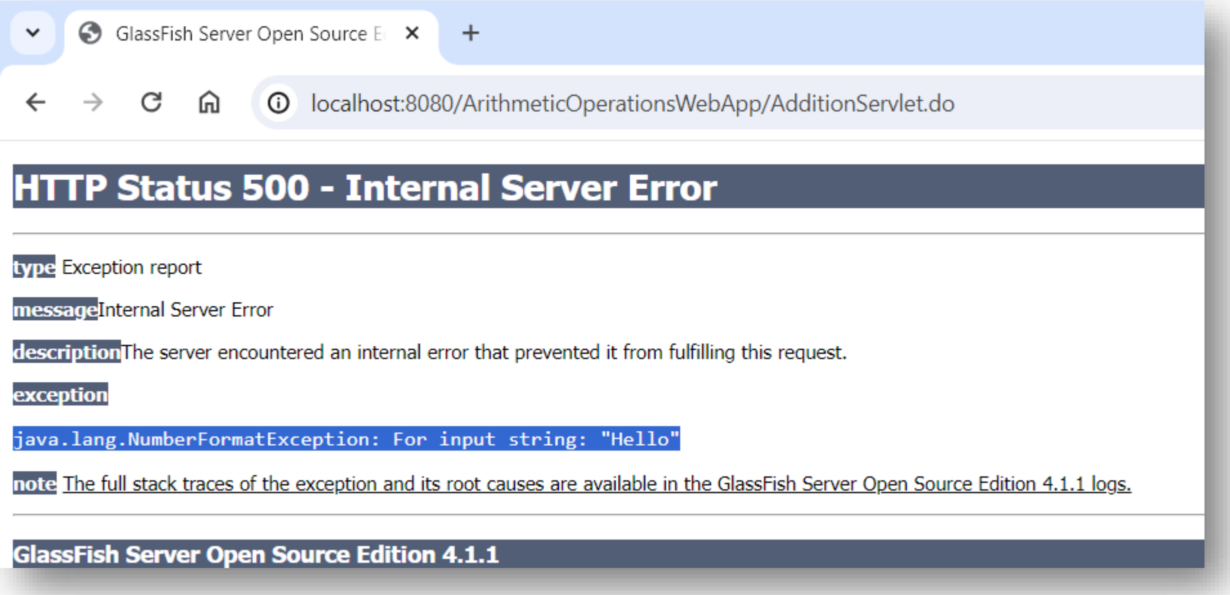
Add numbers

Please enter two numbers below.

First number: 100

Second number: Hello

Click on the ADD button. A **NumberFormatException** is thrown



localhost:8080/ArithmeticOperationsWebApp/AdditionServlet.do

HTTP Status 500 - Internal Server Error

type Exception report

messageInternal Server Error

descriptionThe server encountered an internal error that prevented it from fulfilling this request.

exception

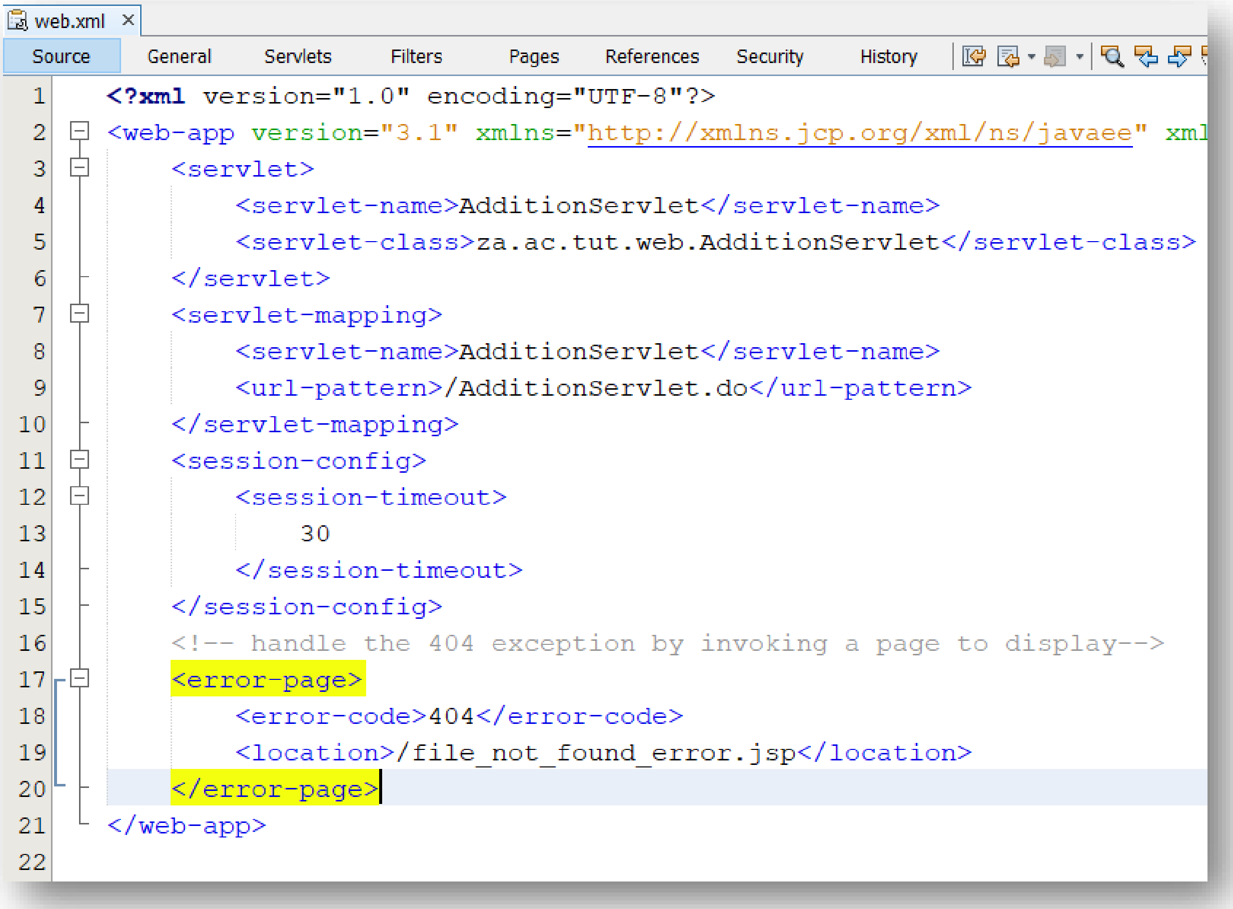
```
java.lang.NumberFormatException: For input string: "Hello"
```

note The full stack traces of the exception and its root causes are available in the GlassFish Server Open Source Edition 4.1.1 logs.

GlassFish Server Open Source Edition 4.1.1

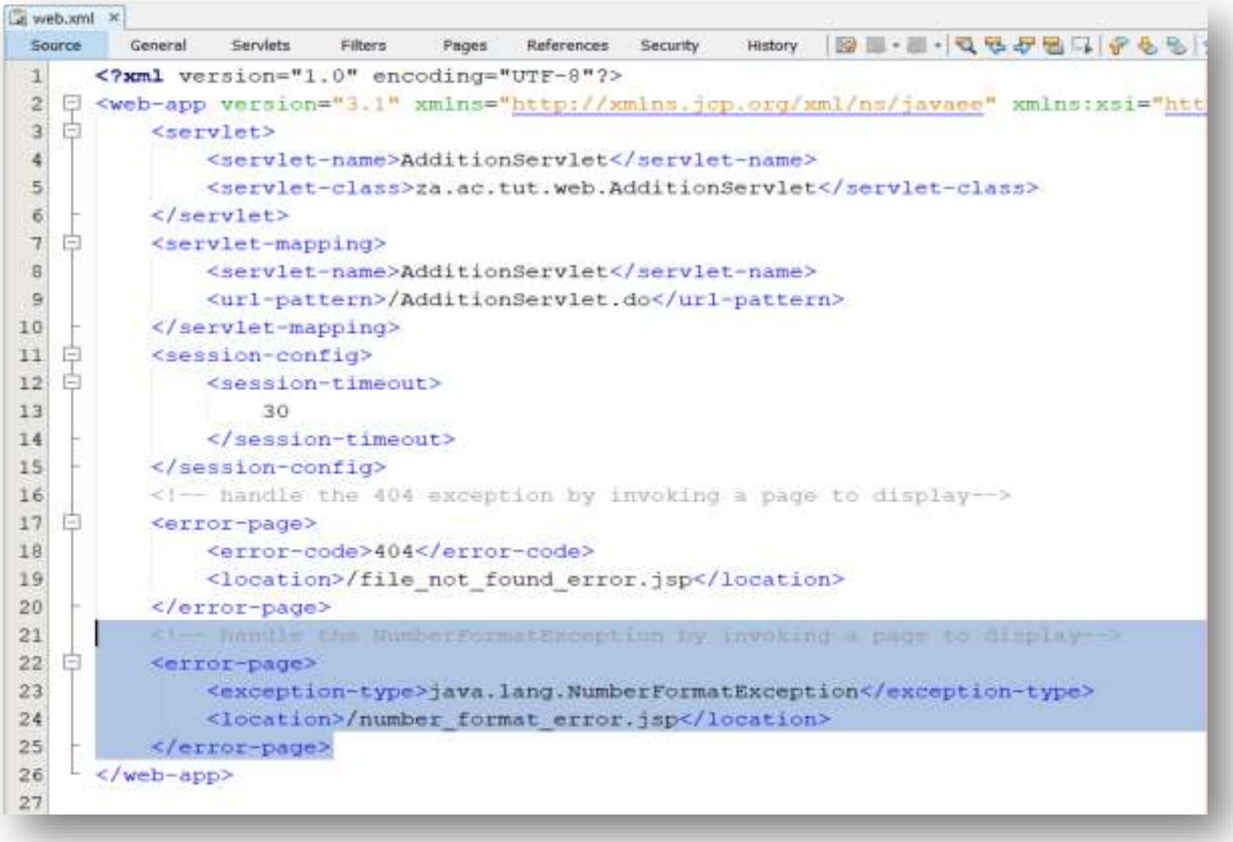
Let’s handle the exception.

- Open the DD file



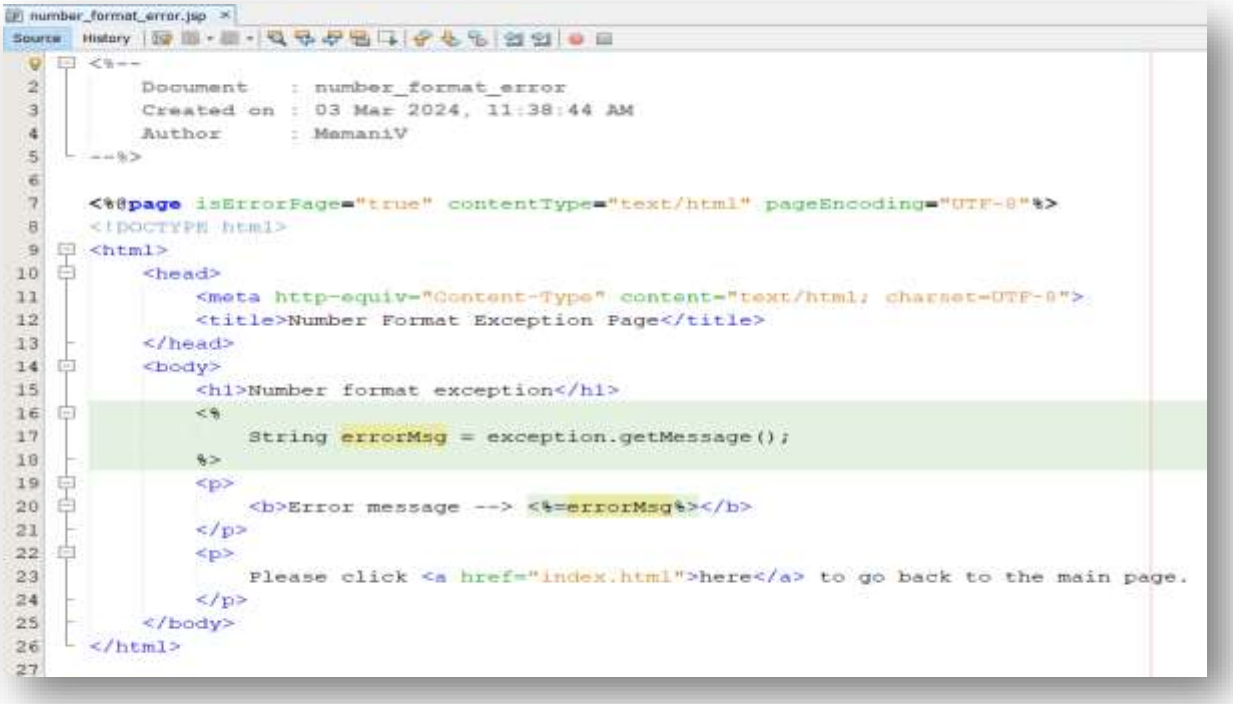
```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xml
    <servlet>
        <servlet-name>AdditionServlet</servlet-name>
        <servlet-class>za.ac.tut.web.AdditionServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>AdditionServlet</servlet-name>
        <url-pattern>/AdditionServlet.do</url-pattern>
    </servlet-mapping>
    <session-config>
        <session-timeout>
            30
        </session-timeout>
    </session-config>
    <!-- handle the 404 exception by invoking a page to display-->
    <error-page>
        <error-code>404</error-code>
        <location>/file_not_found_error.jsp</location>
    </error-page>
</web-app>

```

- State how the NumberFormatException should be handled



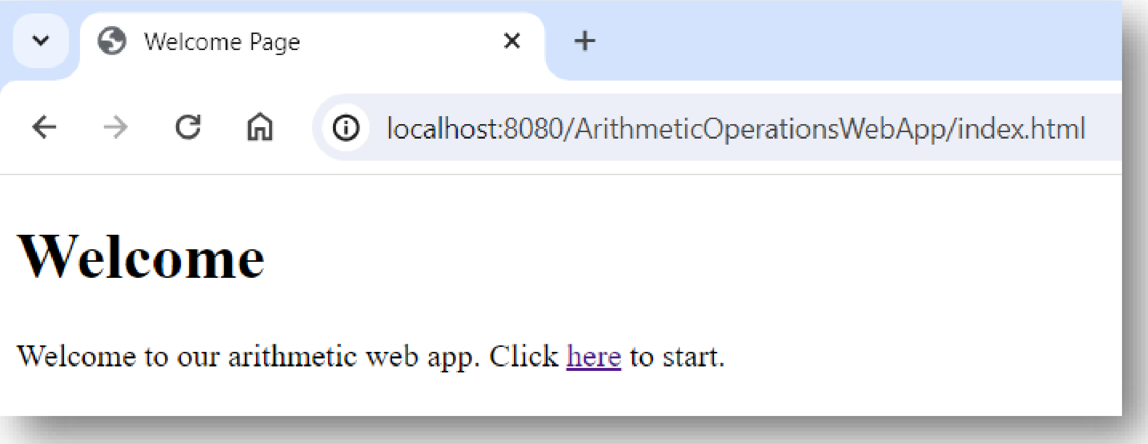
```xml
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xmlns:xsi="htt
    <servlet>
        <servlet-name>AdditionServlet</servlet-name>
        <servlet-class>za.ac.tut.web.AdditionServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>AdditionServlet</servlet-name>
        <url-pattern>/AdditionServlet.do</url-pattern>
    </servlet-mapping>
    <session-config>
        <session-timeout>
            30
        </session-timeout>
    </session-config>
    <!-- handle the 404 exception by invoking a page to display-->
    <error-page>
        <error-code>404</error-code>
        <location>/file_not_found_error.jsp</location>
    </error-page>
    <!-- handle the NumberFormatException by invoking a page to display-->
    <error-page>
        <exception-type>java.lang.NumberFormatException</exception-type>
        <location>/number_format_error.jsp</location>
    </error-page>
</web-app>
```

- Create the **number_format_error.jsp**



```jsp
<%--
    Document   : number_format_error
    Created on : 03 Mar 2024, 11:38:44 AM
    Author     : MemaniV
--%>

<%@page isErrorPage="true" contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Number Format Exception Page</title>
    </head>
    <body>
        <h1>Number format exception</h1>
        <%
            String errorMsg = exception.getMessage();
        %>
        <p>
            <b>Error message --> <%=errorMsg%></b>
        </p>
        <p>
            Please click <a href="index.html">here</a> to go back to the main page.
        </p>
    </body>
</html>
```

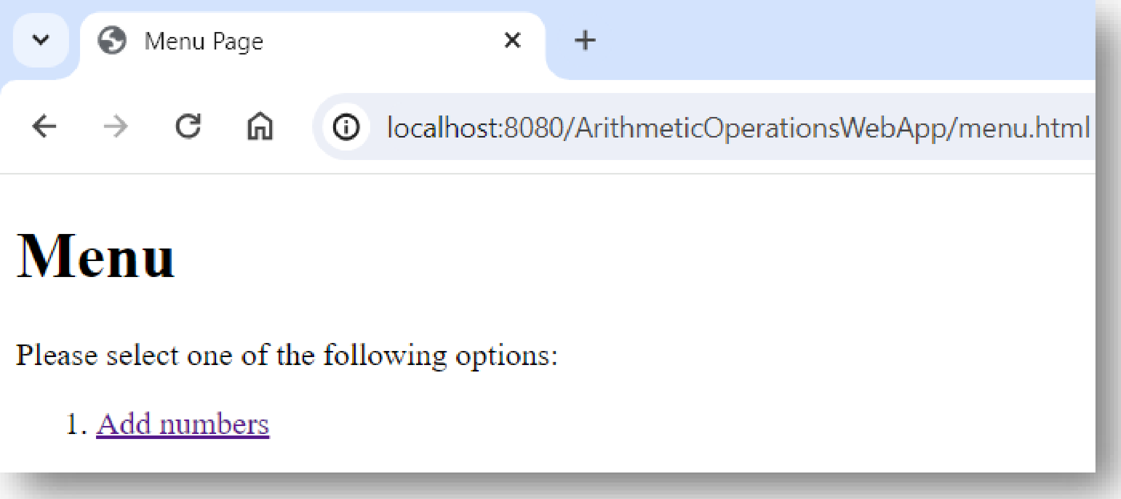
Clean and Build, Deploy and Run



Welcome

Welcome to our arithmetic web app. Click here to start.

Click on the link

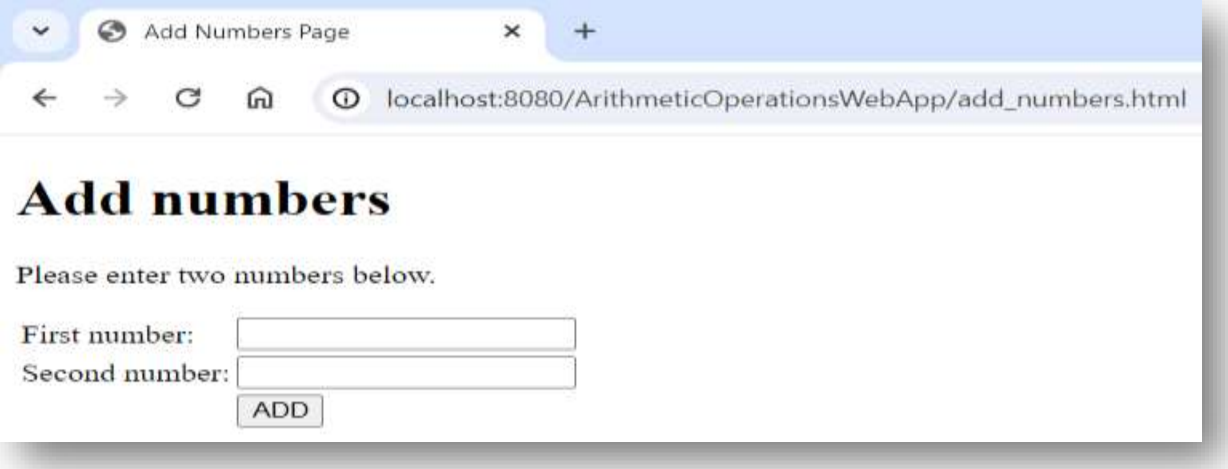


Menu

Please select one of the following options:

1. Add numbers

Click on the link

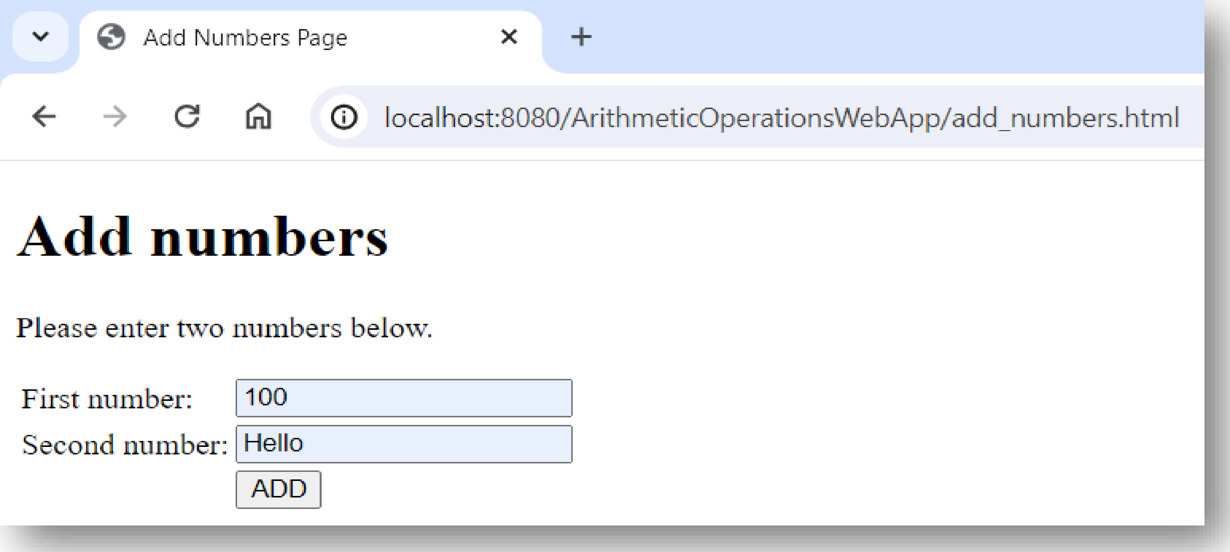


Add numbers

Please enter two numbers below.

First number:	
Second number:	

Enter data

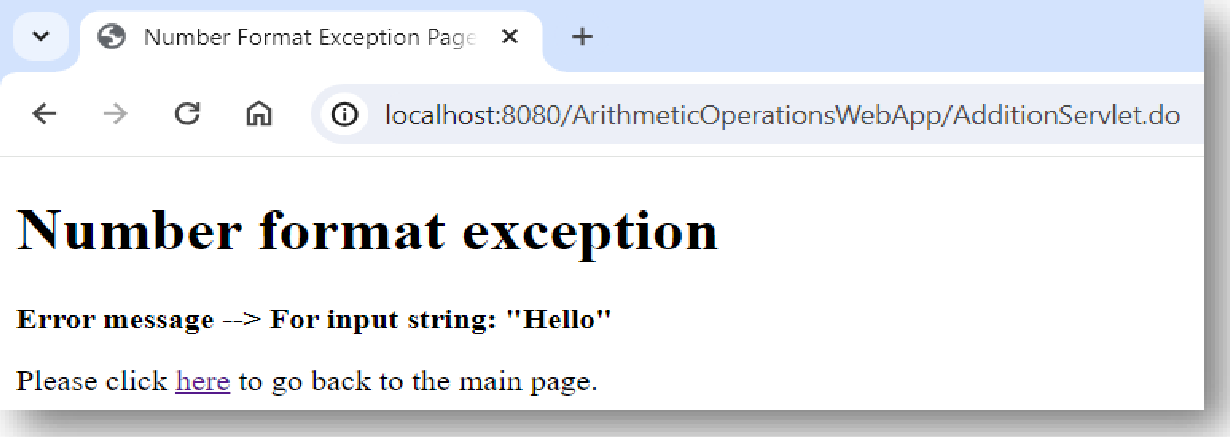


Add numbers

Please enter two numbers below.

First number:	100
Second number:	Hello

Click on the ADD button

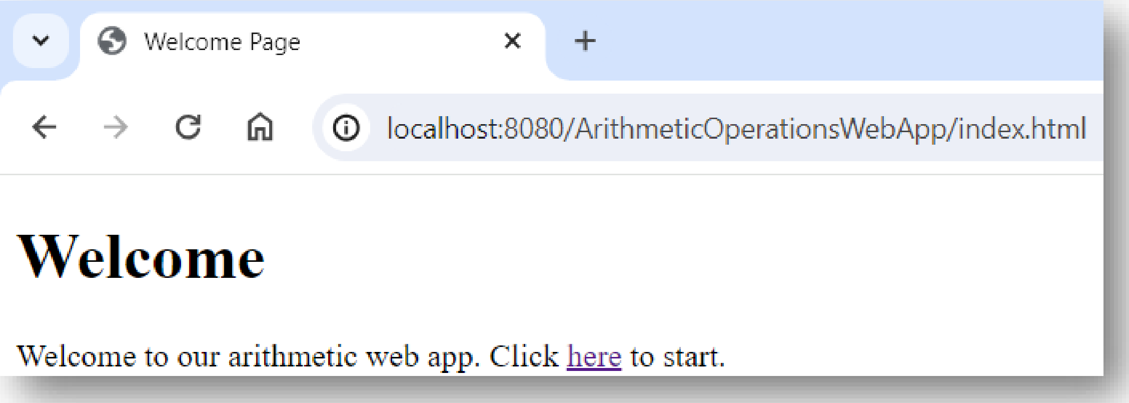

Number Format Exception Page
localhost:8080/ArithmeticOperationsWebApp/AdditionServlet.do

Number format exception

Error message --> For input string: "Hello"

Please click here to go back to the main page.

The exception has been properly handled. Click on the link.

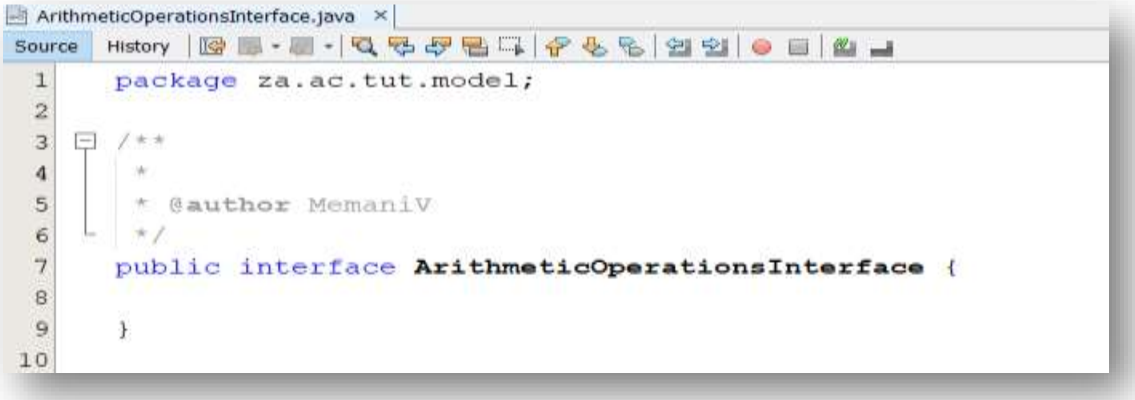


Welcome

Welcome to our arithmetic web app. Click here to start.

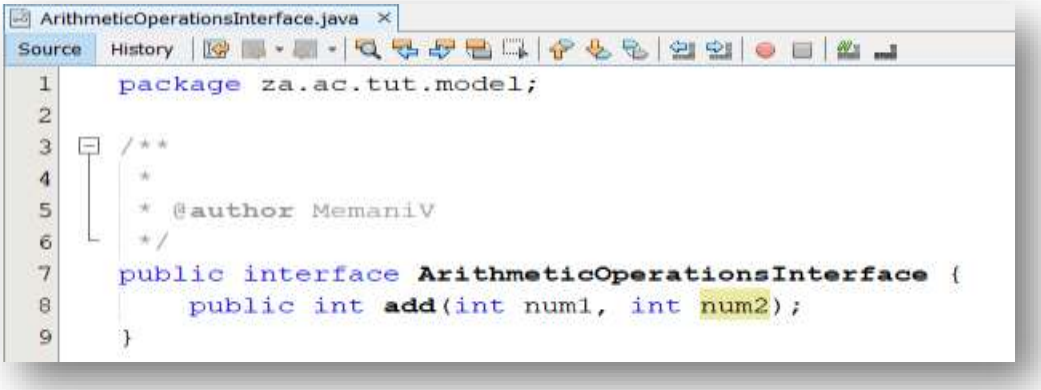
Create business logic.

- Create an interface called **ArithmeticOperationsInterface**



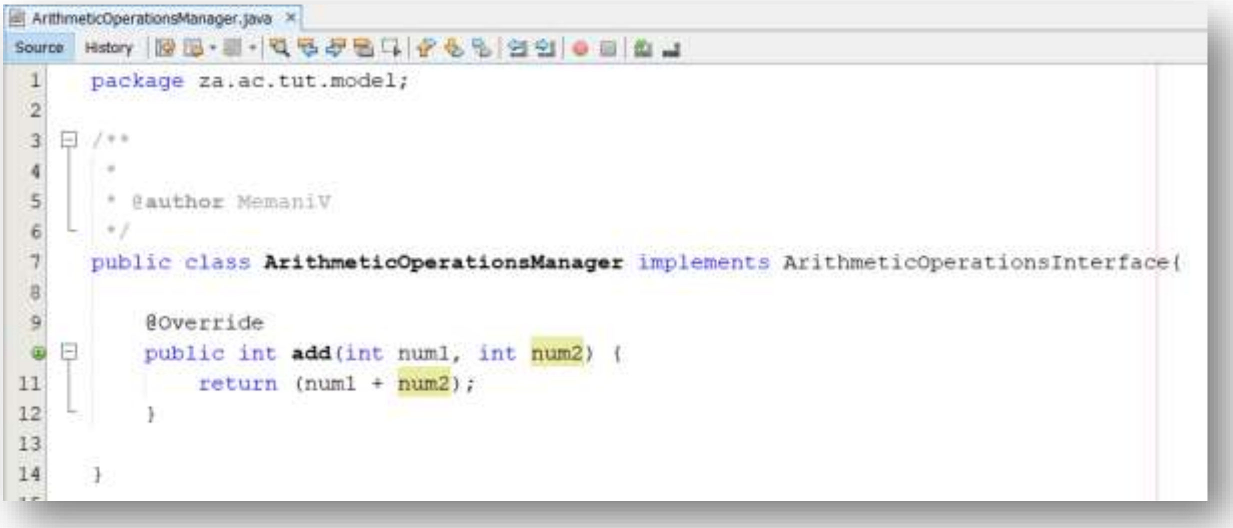
```java
package za.ac.tut.model;

/**
 *
 * @author MemaniV
 */
public interface ArithmeticOperationsInterface {

}

```

- Include a method to add numbers



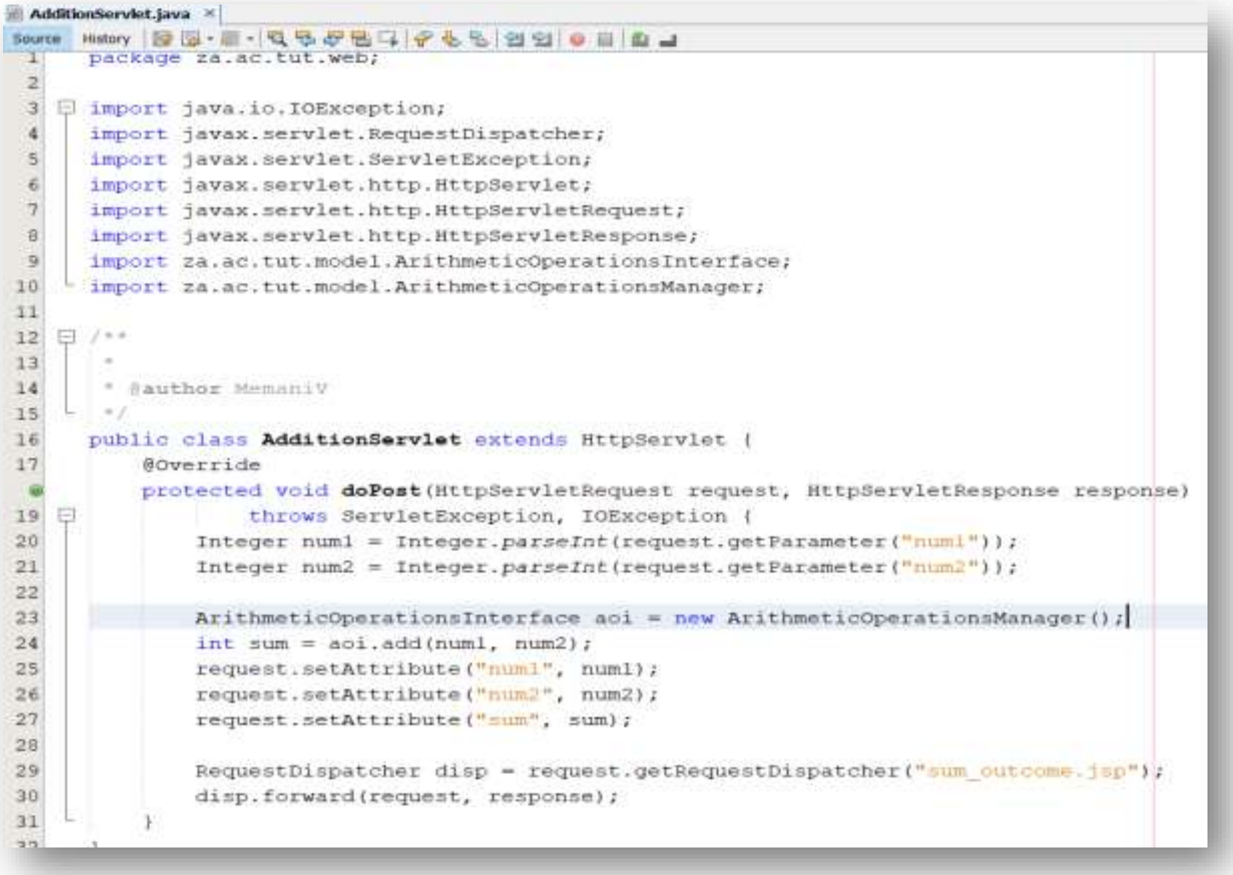
```java
package za.ac.tut.model;

/**
 *
 * @author MemaniV
 */
public interface ArithmeticOperationsInterface {
    public int add(int num1, int num2);
}
```

- Create a class that implements the interface



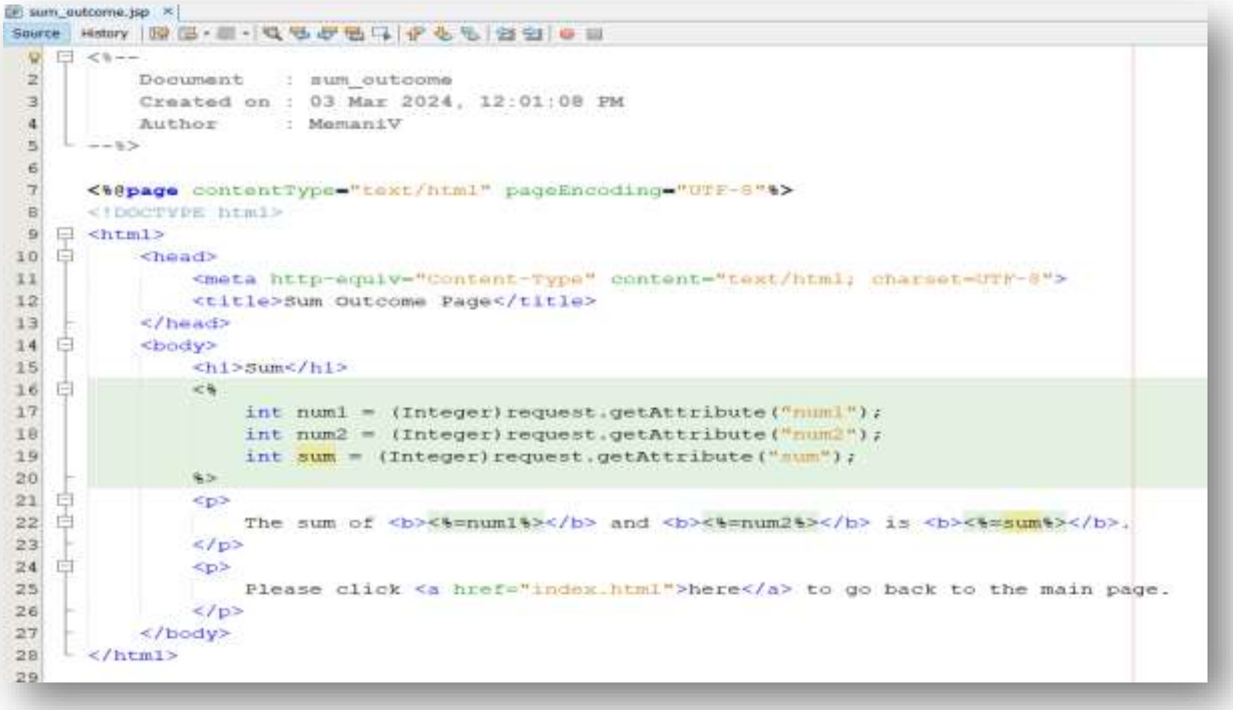
```java
package za.ac.tut.model;

/**
 *
 * @author MemaniV
 */
public class ArithmeticOperationsManager implements ArithmeticOperationsInterface{

    @Override
    public int add(int num1, int num2) {
        return (num1 + num2);
    }

}
```

- Edit the servlet



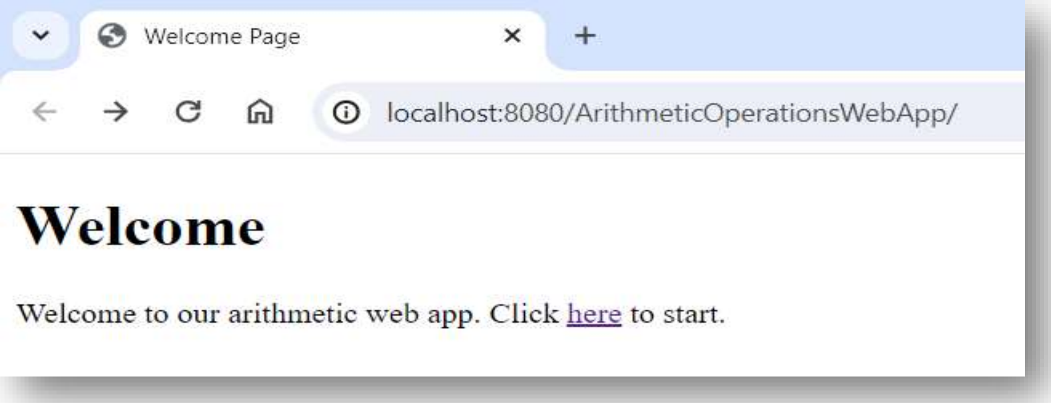
```java
package za.ac.tut.web;

import java.io.IOException;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import za.ac.tut.model.ArithmeticOperationsInterface;
import za.ac.tut.model.ArithmeticOperationsManager;

/**
 *
 * @author MemaniV
 */
public class AdditionServlet extends HttpServlet {
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        Integer num1 = Integer.parseInt(request.getParameter("num1"));
        Integer num2 = Integer.parseInt(request.getParameter("num2"));

        ArithmeticOperationsInterface aoi = new ArithmeticOperationsManager();
        int sum = aoi.add(num1, num2);
        request.setAttribute("num1", num1);
        request.setAttribute("num2", num2);
        request.setAttribute("sum", sum);

        RequestDispatcher disp = request.getRequestDispatcher("sum_outcome.jsp");
        disp.forward(request, response);
    }
}
```

- Create the **sum_outcome.jsp** file



```
<%--
    Document   : sum_outcome
    Created on : 03 Mar 2024, 12:01:08 PM
    Author     : MemaniV
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Sum Outcome Page</title>
    </head>
    <body>
        <h1>Sum</h1>
        <%
            int num1 = (Integer)request.getAttribute("num1");
            int num2 = (Integer)request.getAttribute("num2");
            int sum = (Integer)request.getAttribute("sum");
        %>
        <p>
            The sum of <b><%=num1%></b> and <b><%=num2%></b> is <b><%=sum%></b>.
        </p>
        <p>
            Please click <a href="index.html">here</a> to go back to the main page.
        </p>
    </body>
</html>
```

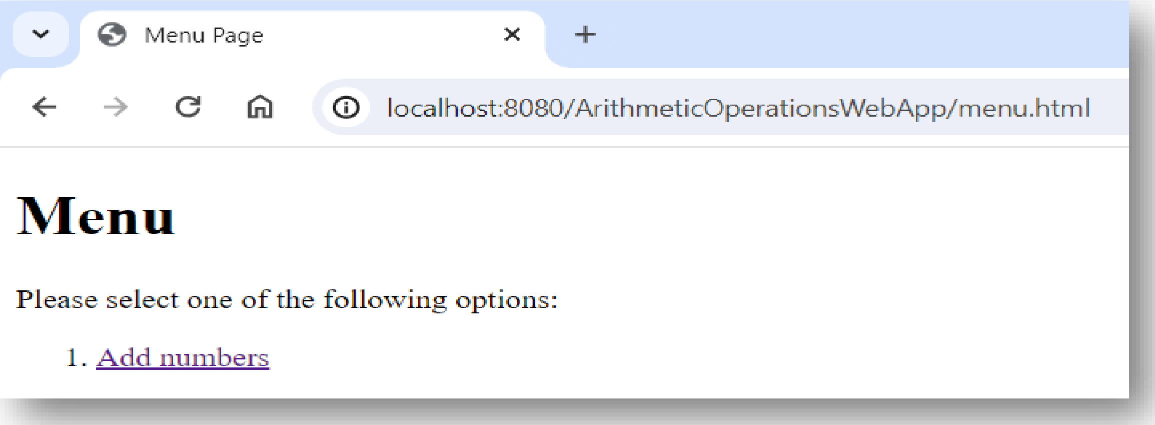
Clean and Build, Deploy and Run.



Welcome

Welcome to our arithmetic web app. Click here to start.

Click on the link

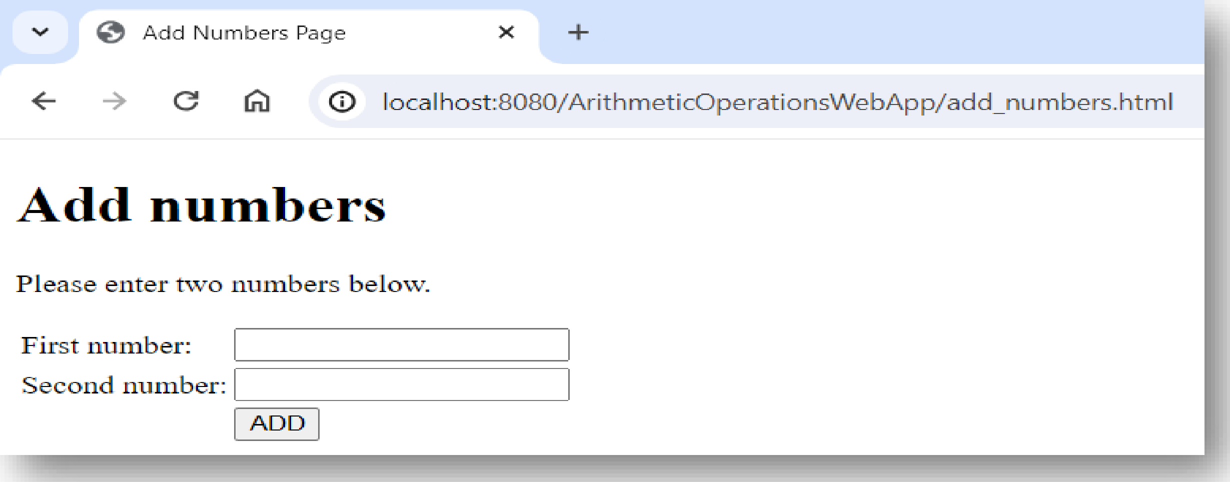


Menu

Please select one of the following options:

1. Add numbers

Click on the link



Add numbers

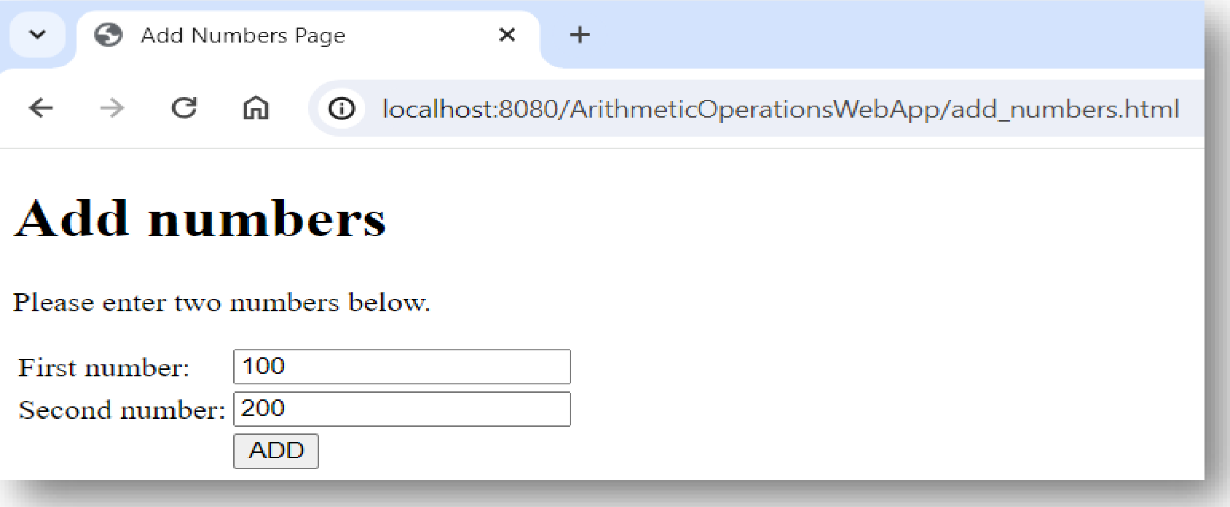
Please enter two numbers below.

First number:

Second number:

ADD

Enter the numbers

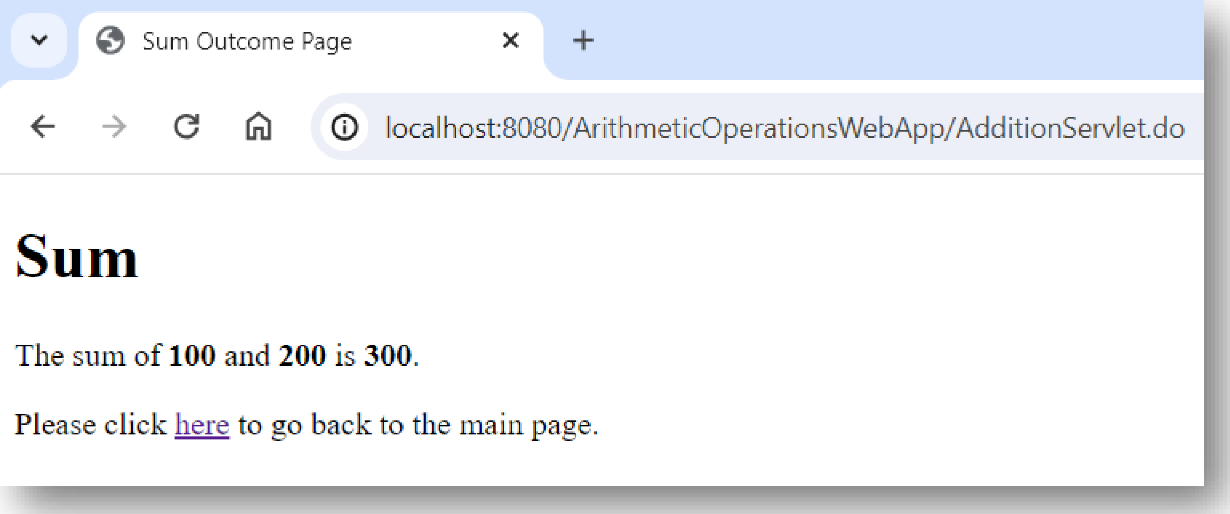


Add numbers

Please enter two numbers below.

First number:	100
Second number:	200
	ADD

Click on the ADD button

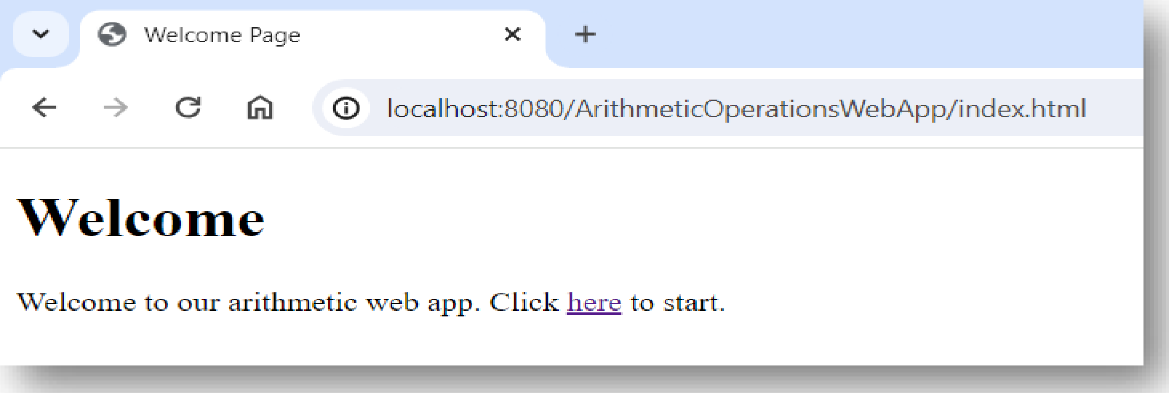

Sum Outcome Page
localhost:8080/ArithmeticOperationsWebApp/AdditionServlet.do

Sum

The sum of **100** and **200** is **300**.

Please click here to go back to the main page.

Click on the link

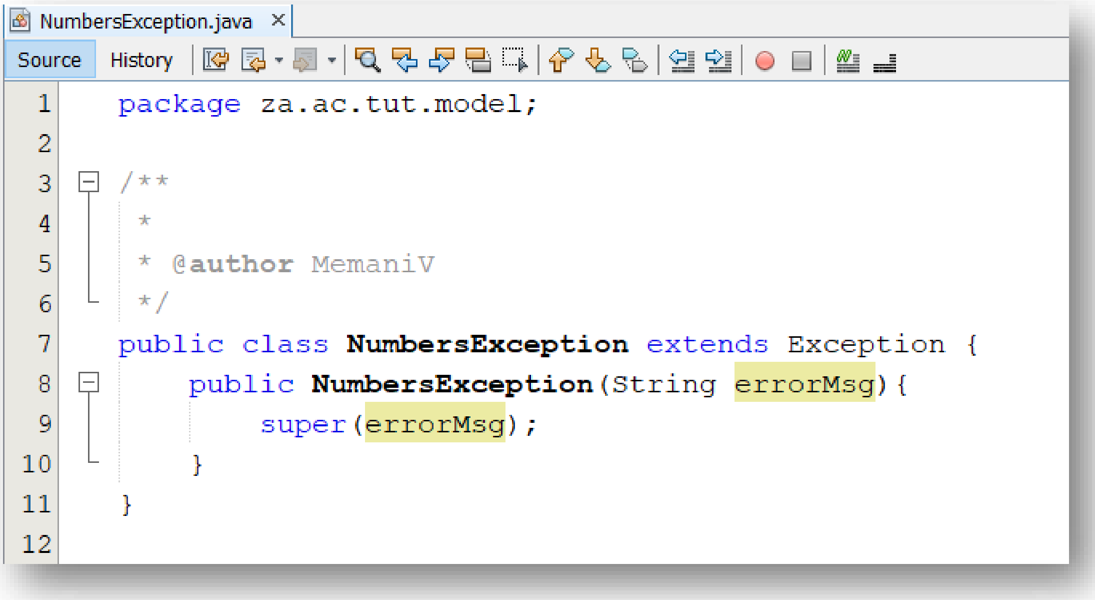


Welcome

Welcome to our arithmetic web app. Click here to start.

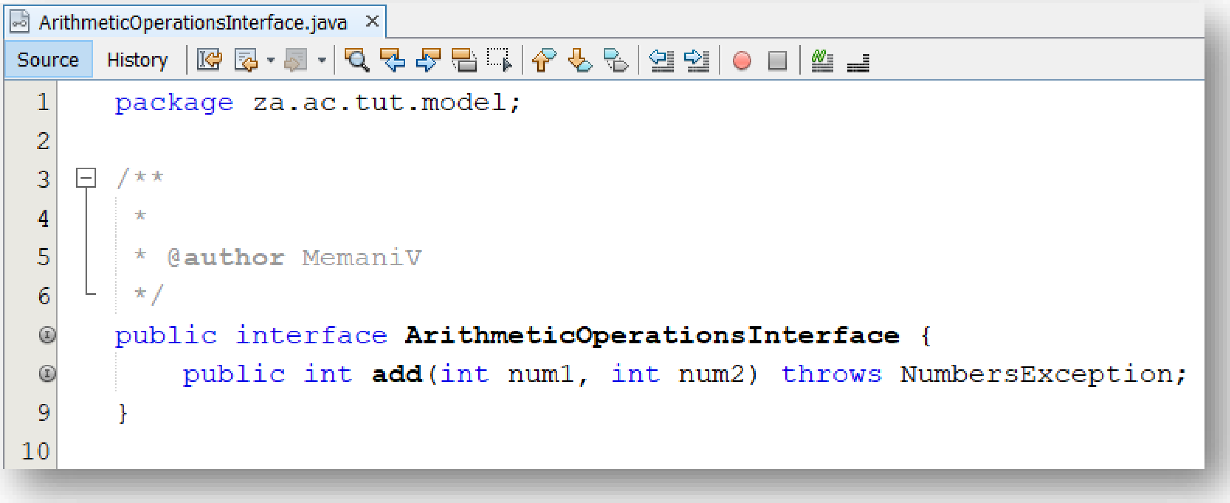
Let’s make another change. Let the application only work with positive integers. This means we have to define our own exception class.

- Create an exception class called **NumbersException**



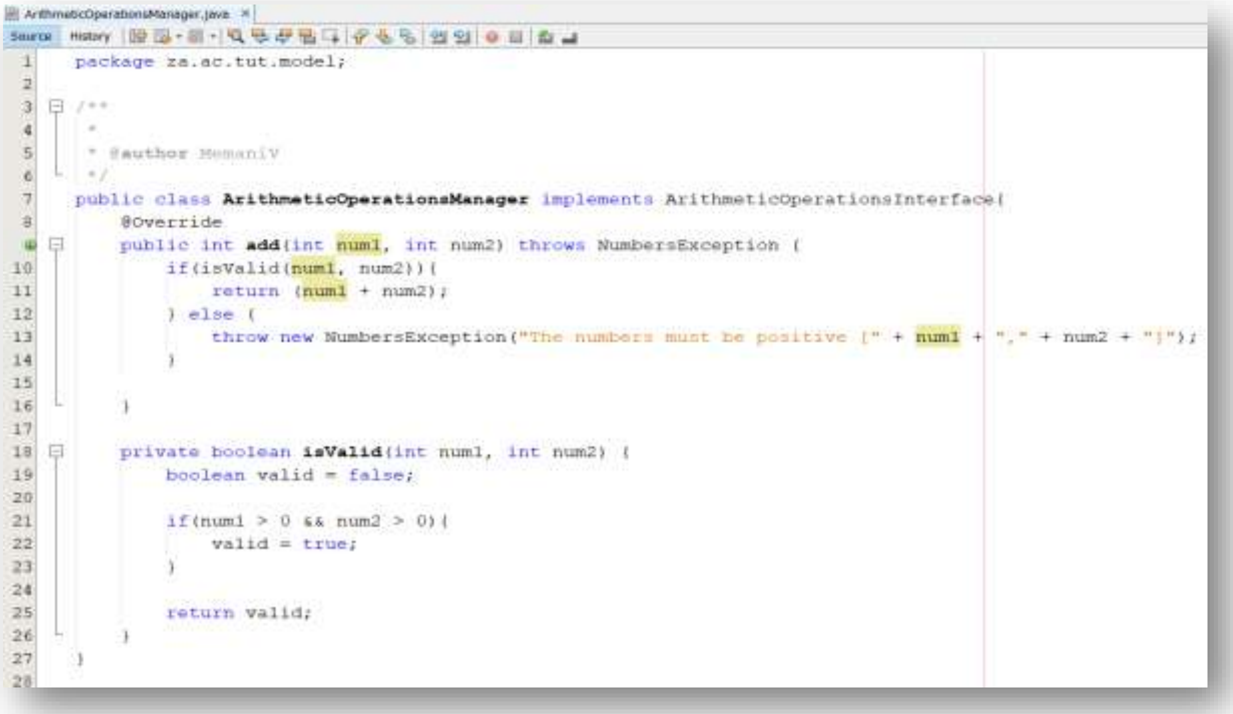
```java
package za.ac.tut.model;

/**
 *
 * @author MemaniV
 */
public class NumbersException extends Exception {
    public NumbersException(String errorMsg){
        super(errorMsg);
    }
}

```

- Modify the method header of the interface method to include the NumbersException



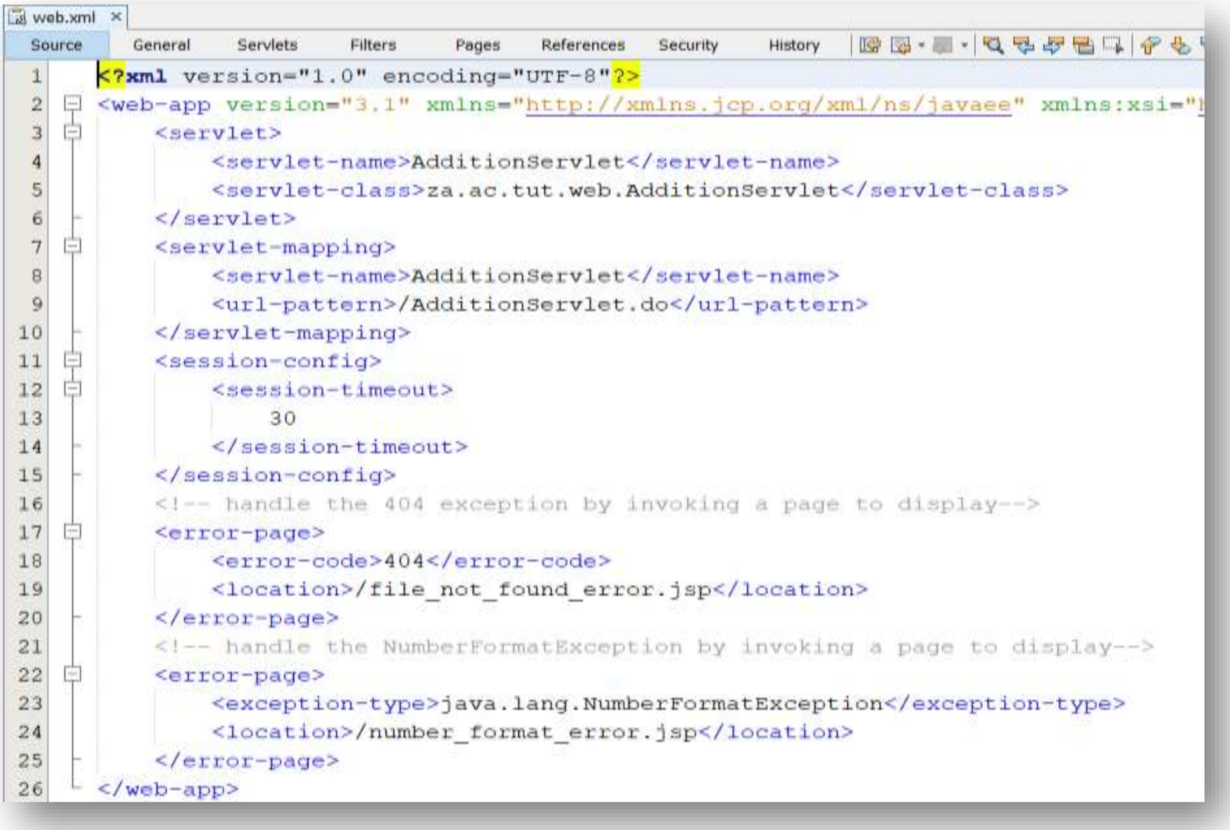
```java
package za.ac.tut.model;

/**
 *
 * @author MemaniV
 */
public interface ArithmeticOperationsInterface {
    public int add(int num1, int num2) throws NumbersException;
}

```

- Modify the manager class



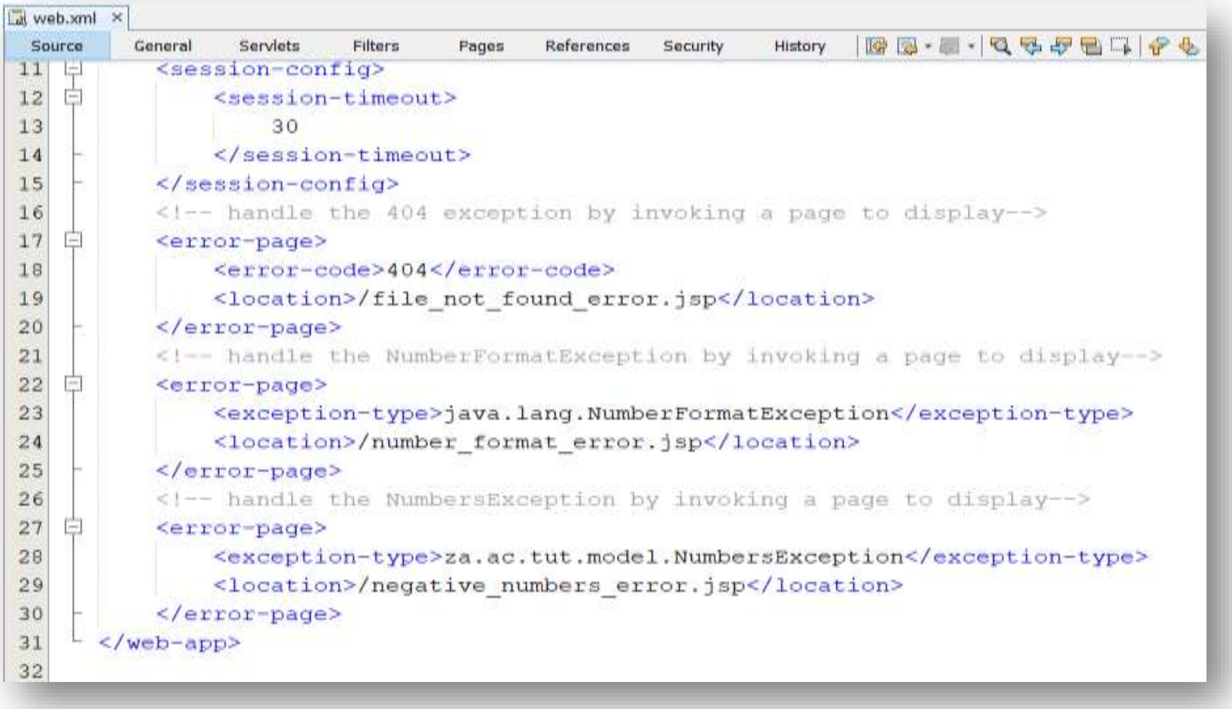
```java
package za.ac.tut.model;

/**
 *
 * @author MemaniV
 */
public class ArithmeticOperationsManager implements ArithmeticOperationsInterface{
    @Override
    public int add(int num1, int num2) throws NumbersException {
        if(isValid(num1, num2)){
            return (num1 + num2);
        } else {
            throw new NumbersException("The numbers must be positive [" + num1 + "," + num2 + "]");
        }

    }

    private boolean isValid(int num1, int num2) {
        boolean valid = false;

        if(num1 > 0 && num2 > 0){
            valid = true;
        }

        return valid;
    }
}
```

- Open the DD file



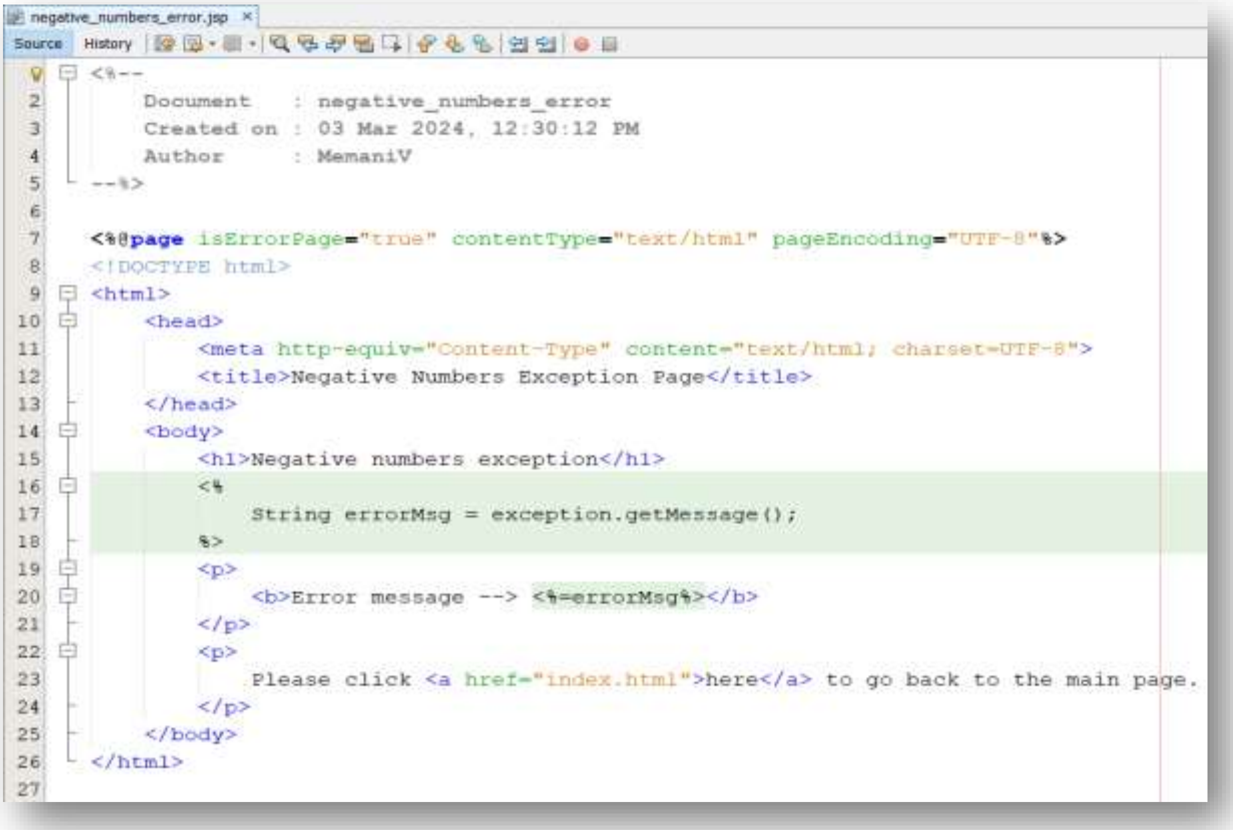
```xml
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xmlns:xsi="
    <servlet>
        <servlet-name>AdditionServlet</servlet-name>
        <servlet-class>za.ac.tut.web.AdditionServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>AdditionServlet</servlet-name>
        <url-pattern>/AdditionServlet.do</url-pattern>
    </servlet-mapping>
    <session-config>
        <session-timeout>
            30
        </session-timeout>
    </session-config>
    <!-- handle the 404 exception by invoking a page to display-->
    <error-page>
        <error-code>404</error-code>
        <location>/file_not_found_error.jsp</location>
    </error-page>
    <!-- handle the NumberFormatException by invoking a page to display-->
    <error-page>
        <exception-type>java.lang.NumberFormatException</exception-type>
        <location>/number_format_error.jsp</location>
    </error-page>
</web-app>
```

- Include the **za.ac.tut.model.NumbersException**



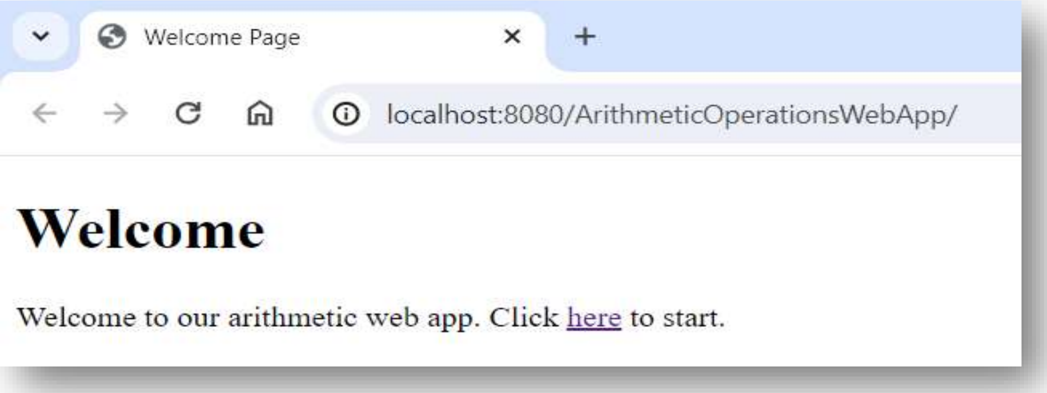
```
    <session-config>
        <session-timeout>
            30
        </session-timeout>
    </session-config>
    <!-- handle the 404 exception by invoking a page to display-->
    <error-page>
        <error-code>404</error-code>
        <location>/file_not_found_error.jsp</location>
    </error-page>
    <!-- handle the NumberFormatException by invoking a page to display-->
    <error-page>
        <exception-type>java.lang.NumberFormatException</exception-type>
        <location>/number_format_error.jsp</location>
    </error-page>
    <!-- handle the NumbersException by invoking a page to display-->
    <error-page>
        <exception-type>za.ac.tut.model.NumbersException</exception-type>
        <location>/negative_numbers_error.jsp</location>
    </error-page>
</web-app>
```

- Create the **negative_numbers_error.jsp** file



```
<%--
    Document   : negative_numbers_error
    Created on : 03 Mar 2024, 12:30:12 PM
    Author     : MemaniV
--%>

<%@page isErrorPage="true" contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Negative Numbers Exception Page</title>
    </head>
    <body>
        <h1>Negative numbers exception</h1>
        <%
            String errorMsg = exception.getMessage();
        %>
        <p>
            <b>Error message --> <%=errorMsg%></b>
        </p>
        <p>
            Please click <a href="index.html">here</a> to go back to the main page.
        </p>
    </body>
</html>
```

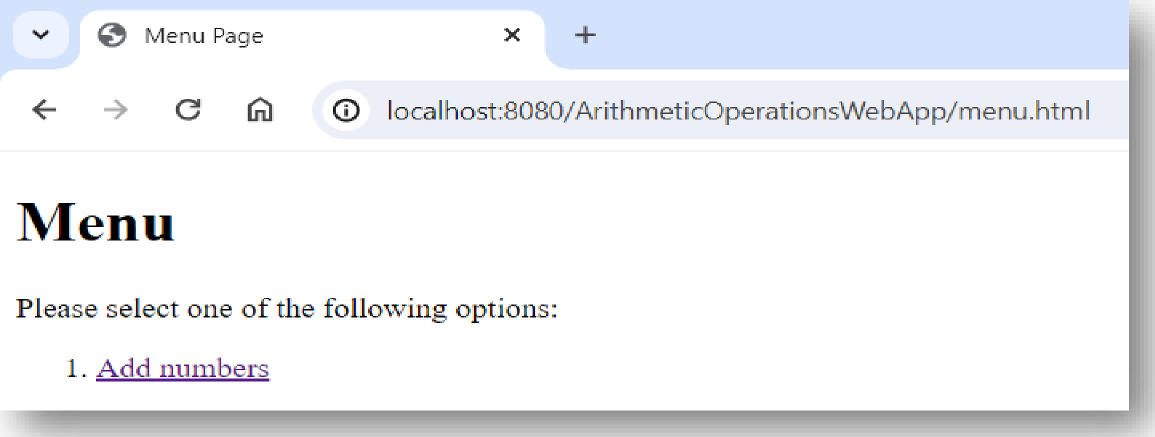
Clean and Build, Deploy and Run.



Welcome

Welcome to our arithmetic web app. Click here to start.

Click on the link

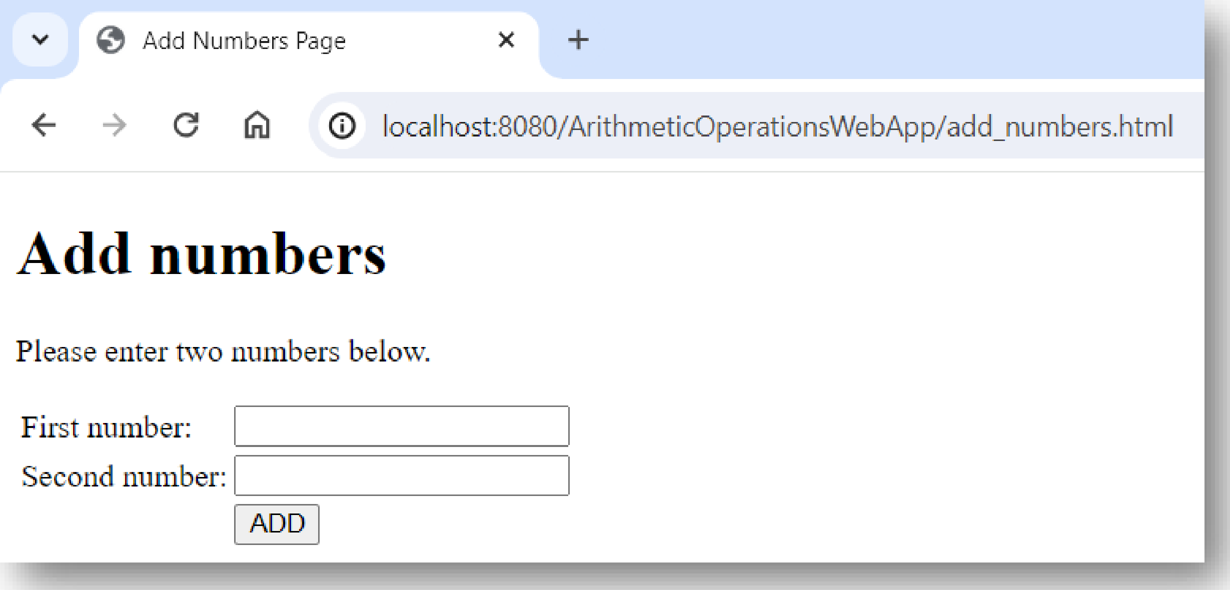


Menu

Please select one of the following options:

1. Add numbers

Click on the link



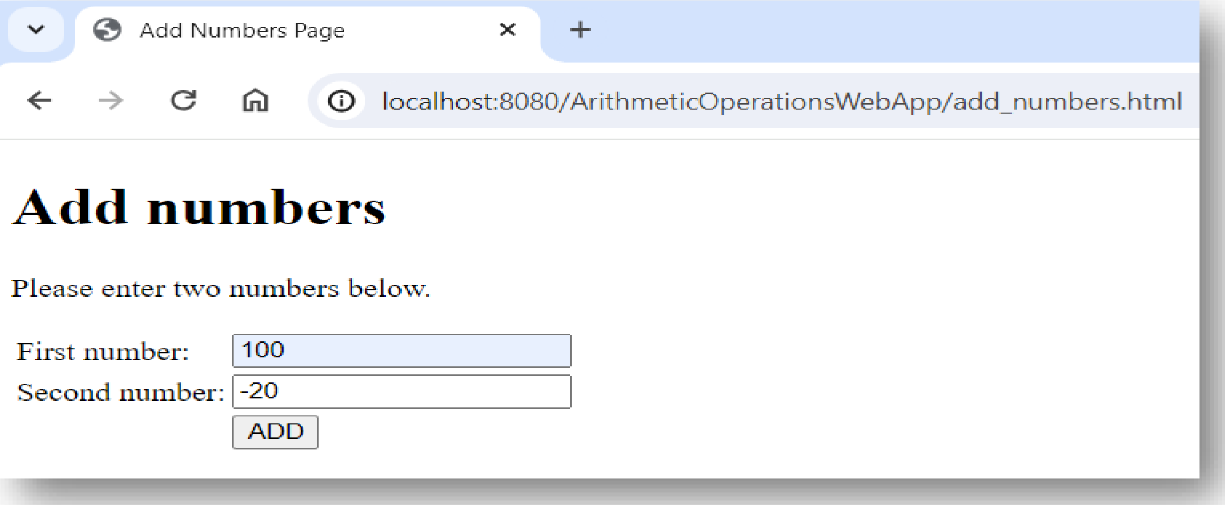
Add numbers

Please enter two numbers below.

First number:

Second number:

Enter the numbers



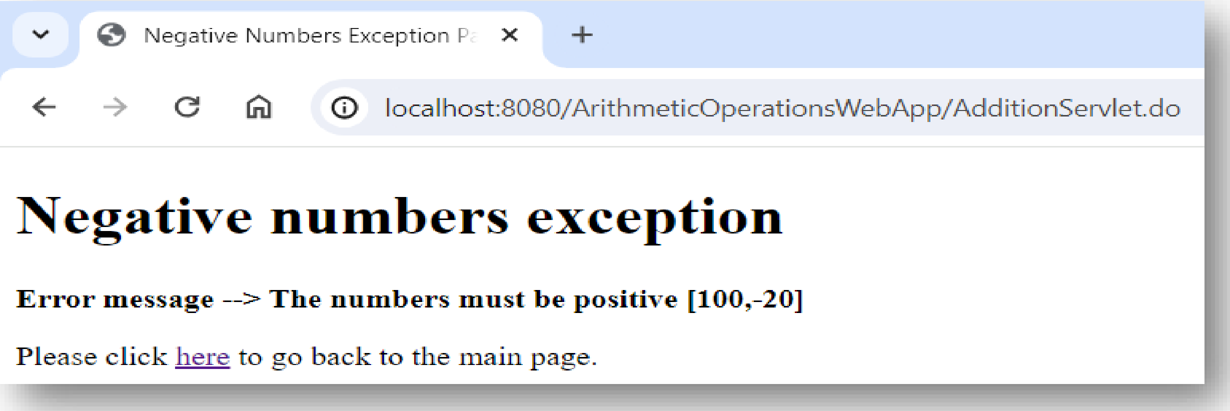
Add numbers

Please enter two numbers below.

First number: 100

Second number: -20

Click on the ADD button

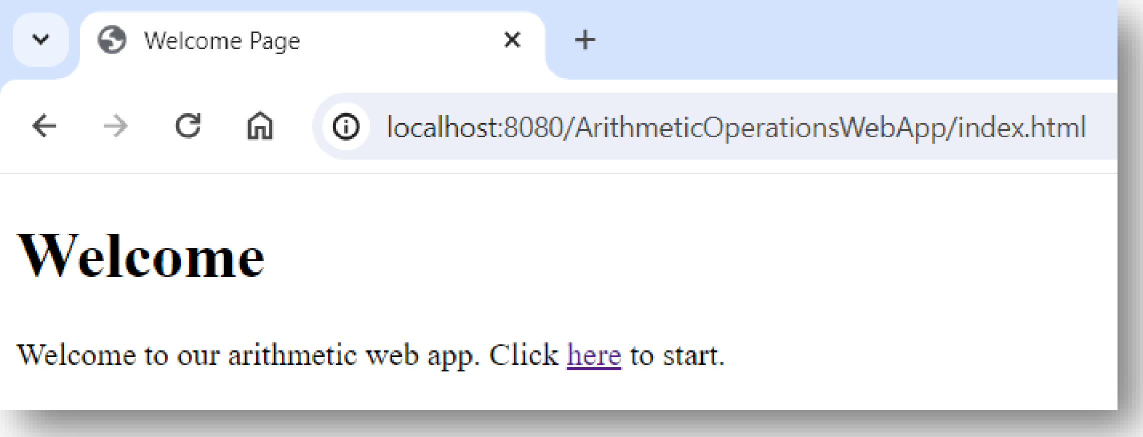

Negative Numbers Exception P
localhost:8080/ArithmeticOperationsWebApp/AdditionServlet.do

Negative numbers exception

Error message --> The numbers must be positive [100,-20]

Please click here to go back to the main page.

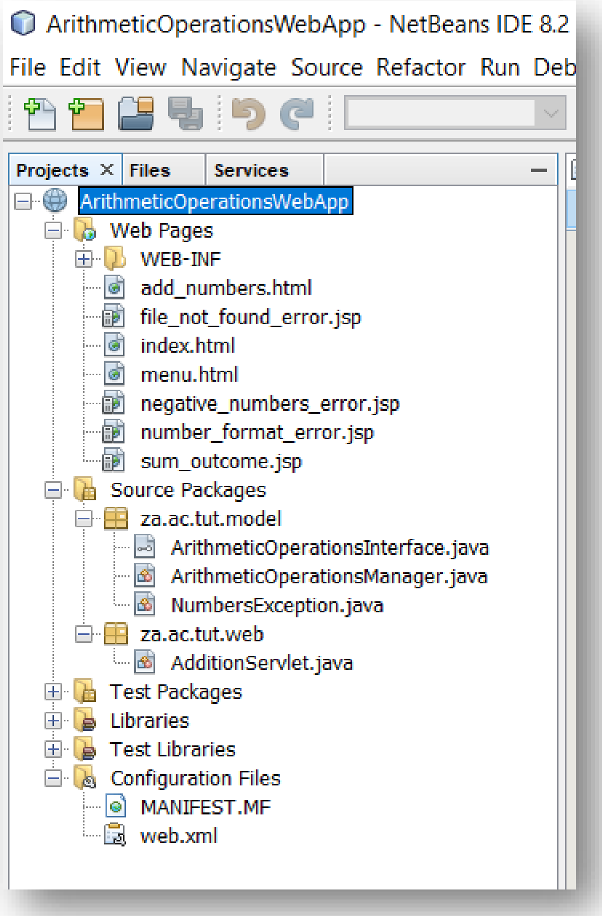
Click on the link



Welcome

Welcome to our arithmetic web app. Click here to start.

Complete project structure



ArithmeticOperationsWebApp - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Deb
Projects
Files
Services

```
ArithmeticOperationsWebApp
    Web Pages
        WEB-INF
        add_numbers.html
        file_not_found_error.jsp
        index.html
        menu.html
        negative_numbers_error.jsp
        number_format_error.jsp
        sum_outcome.jsp
    Source Packages
        za.ac.tut.model
            ArithmeticOperationsInterface.java
            ArithmeticOperationsManager.java
            NumbersException.java
        za.ac.tut.web
            AdditionServlet.java
    Test Packages
    Libraries
    Test Libraries
    Configuration Files
        MANIFEST.MF
        web.xml
```

1.7 DIY (Do It Yourself)

In this DIY we want you to develop robust web applications through exception handling.

Task #1

Modify the example and add three more arithmetic operations (subtraction, multiplication and division).

Task #2

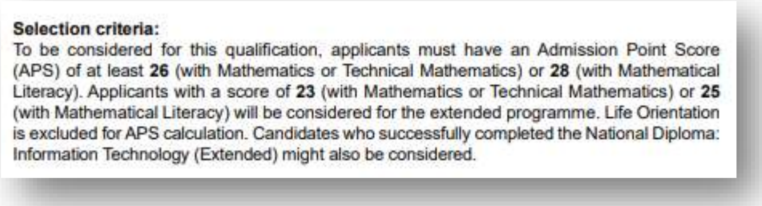
Relebogile is an intern student at ***CodersThink.Inc*** company. The company specialises in developing web applications for companies. As his first task at the company, Relebogile is given the responsibilty of creating a web application that will help users come up with strong passwords. A strong password must meet the following requirements:

- The size must be at least 10 keys.
- Consist of letters, special characters and numbers. At least two of the letters must be uppercase, and another two lowercase.
- The @ and # special characters must not be part of the password.

Assuming that you are Relebogile, create such a web application for ***CodersThink.Inc***.

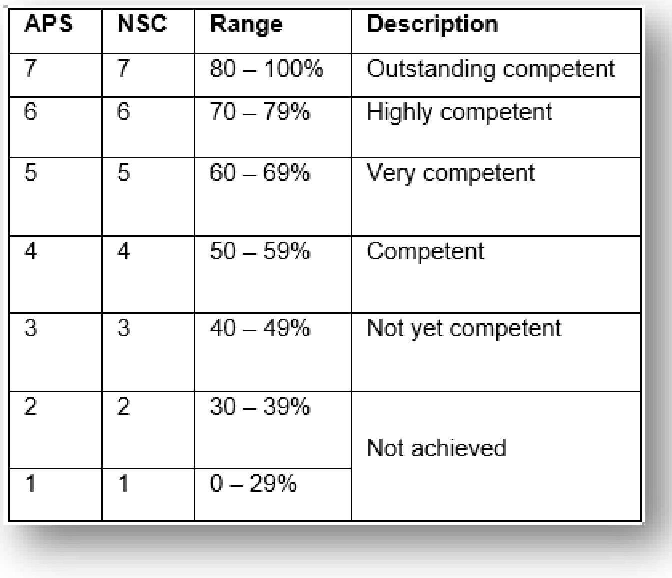
Task #3

Mulumba is a newly appointed software developer at an academic institution, XYZ. The institution has a selection criteria for first year intake into their Computer Science programme. Below is the selection criteria.



Selection criteria:
To be considered for this qualification, applicants must have an Admission Point Score (APS) of at least **26** (with Mathematics or Technical Mathematics) or **28** (with Mathematical Literacy). Applicants with a score of **23** (with Mathematics or Technical Mathematics) or **25** (with Mathematical Literacy) will be considered for the extended programme. Life Orientation is excluded for APS calculation. Candidates who successfully completed the National Diploma: Information Technology (Extended) might also be considered.

The institution wants a web application that prospective students can use to calculate their APS (Admission Point Score) before applying. This will save students time and money so that they don't apply if they don't meet the minimum requirements. The table below shows the relationship between APS and the grades obtained by students at matric (NSC).



APS	NSC	Range	Description
7	7	80 – 100%	Outstanding competent
6	6	70 – 79%	Highly competent
5	5	60 – 69%	Very competent
4	4	50 – 59%	Competent
3	3	40 – 49%	Not yet competent
2	2	30 – 39%	Not achieved
1	1	0 – 29%	

So the web application must allow a prospective student to enter his/her NSC grades, together with the respective subject names. The application must then calculate the APS and tell whether the student qualifies or not. The application must make sure that the NSC grades are in the specified ranges. Also Life Orientation information must not be accepted.

Assuming that you are Mulumba, create such a web application for institution XYZ.

1.8 Conclusion

In this chapter we managed to introduce the student to exceptions and their handling. In the next chapter we discuss the concept of EJBs, the business component of JEE.

Thank you very much for having taken time to go through this chapter. Enjoy the rest of the day and God bless you.