Date: _____ **Course/Class:** _____

Student ID: _____ **Student Name:** _____

Higher Diploma in Software Engineering (IT114105)

Enterprise Systems Development (ITP4511)

Test One

Total marks: 60 marks

Time Allowed: 60 Minutes

Q1 A simple web application is developed with Java Server Page. It shows an image of a medal based on the value of an input mark. The application is composed of `medalDB.jsp`, `q1.jsp` and `q1Handler.jsp`.

- The `medalDB.jsp` defines a `String[] getAllMedals` method in JSP declaration. The method returns all image locations in String array.
- The `q1.jsp` shows images of medals in table format and an input form for user submission. The locations of medal images are specified in the `getAllMedals` method.
- The `q1Handler.jsp` obtains input parameter, `mark`, validates input parameter and displays the corresponding value of the mark. It also displays a corresponding medal image if the mark is within a specific range. The mark range is between 50 to 100 inclusively. If user does not submit a parameter, `mark`, then an error message will be displayed.

Figure Q1a. A sample output of the q1.jsp

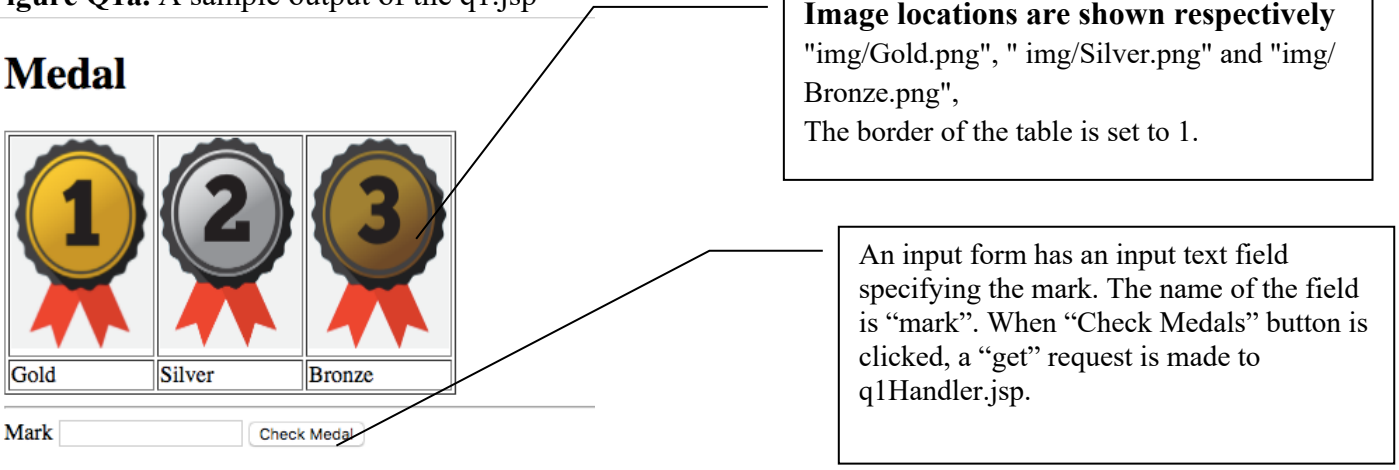
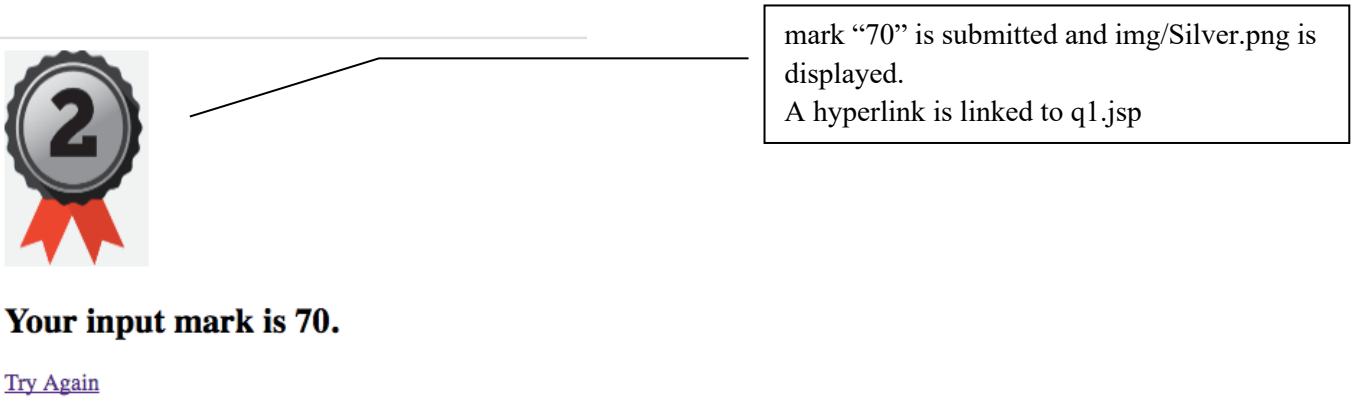


Figure Q1b. A sample output of the URL:
http://localhost:8080/1718_ESD_t1/q1Handler.jsp?mark=70&submit=Check+Medal



The code fragment of **medalDB.jsp** is shown below:

```
<!--
 part (a) (i)
-->
```

(a) In `medalDB.jsp`, write **JSP Declaration** to define a public `String[] getAllMedals()` method that returns all image locations in String array. The image locations are "img/Gold.png", " img/Silver.png" and "img/ Bronze.png" respectively [3%]

The code fragment of **q1.jsp** is shown below:

```
<!--   part (b) (i) -->
<!DOCTYPE html> <html><head><title>q1</title> </head> <body>
  <h1>Medals </h1>
  <table border="1">
    <!--   part (b) (ii) -->
    <tr> <td> Gold </td> <td> Silver </td> <td> Bronze </td> </tr>
  </table>
  <hr/>
  <!-- part (b) (iii) -->
</body></html>
```

(b) With reference to the sample output of the `q1.jsp`, write code to complete the following tasks.

(i) Write a **JSP Directive** to include the `medalDB.jsp`. [2%]

(ii) Write **JSP Scriptlet** to output the table as per the Figure Q1a. You are **MUST** use the `getAllMedals` () method. [5%]

(iii) Write **HTML code** to show an input form. [3%]

The code fragment of q1Handler.jsp is shown below:

```
<!--   Assume the medalDB.jsp is properly included -->
<!DOCTYPE html>
<html><head><title>q1Handler</title> </head>
  <body>
     <%   String input = ""; int mark = 0;
         //  part (c) (i), (ii), (iii)
     %>
     <h2> Your input mark is <!--   part (c) (iv) -->  </h2>
     <!--   part (c) (v) -->
  </body>
</html>
```

(c) With reference to the `q1Handler.jsp` and the figure Q1b, complete the following task:

(i) Obtain the parameter, `mark`, and assign it to `input`. [1%]

(ii) If the parameter, `input`, is not submitted, output the following message "You did not submit any value!!" [2%]

(iii) If the parameter, `input` is submitted, parse `input` to an integer and assign it to the variable `mark`. Use the `getAllMedals()` method and output corresponding image. The range of the medals defined as follows. [5%]

Gold	Between 75 to 100 (inclusive)
Silver	Between 60 to 74 (inclusive)
Bronze	Between 50 to 59 (inclusive)

(iv) Use **JSP Expression** to display the value of the mark. [1%]

(v) Write **HTML** to display a hyperlink linked to q1.jsp for trying again. [1%]

(vi) Write **JSP Directive** in `q1Handler.jsp` to pass the exception to `handleError.jsp`. Assume the `handleError.jsp` is already defined. [2%]

Q2 A web application built with the technologies, JSP, JavaBeans and Servlet for displaying product information.

- The JSP page, `enquiry.jsp`, is an input form. It allows multiple selection for product codes.
- The Java servlet class, `ict.servlet.ProductController`, handles inputted parameter, obtains the corresponding product information from database and displays it accordingly.
- The class, `ict.db.ProductDB`, provides methods to handle the database operations.
- A JavaBean class, `ict.bean.Product`, is used to support the form operations

Figure Q2a. A sample output of *enquiry.jsp*.

Title is in heading 2.
An input form has an input field, **code**.
The product codes are p1, p2 and p3.
The **post** method is used in the form.
When "Submit" button is clicked, the form is submitted to a request to productInfo, which mapped to the ict.servlet.ProductController.java

Figure Q2b. A sample output of the servlet.

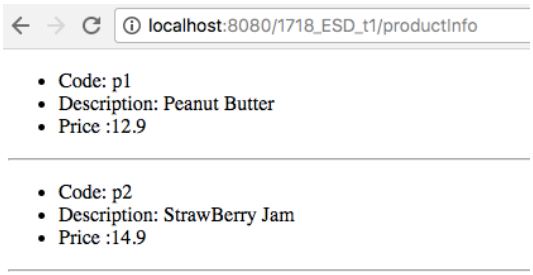


Figure Q2c. The API of ict.db.ProductDB.java.

```
// It returns a Product of given id
public Product getProductByCode (String code)
// It add a Product to the database
public boolean addProduct(Product p)
```

Figure Q2d. Partial source code of *enquiry*.jsp.

```
<html> <head>  <title> </title> </head> <body>
  (a)
</body></html>
```

Figure Q2e. Partial source code of ict.bean.Product.java.

```
package ict.bean;
import java.io.Serializable;
public class Product implements Serializable{
   public Product(String code, String desc, double price) {
     /* intentionally omitted*/
   }
   // (b)
}
```

(a) Write down the well-formed HTML in the `enquiry.jsp` in the Figure Q2d to display the sample output as per the **Figure Q2a**. [4%]

(b) Write code for completing the implementation of the java bean class `ict.bean.Product` class to support the web application development. [6%]

(c) Refer to the sample output shown in the **Figure Q2b** and write code to complete the implementation of the servlet. [15%]

(i) Map the request, `productInfo`, with the name, `controller`, to the `ict.servlet.ProductController.java`.

(ii) Generalize the class to support servlet implementation.

(iii) State the name of the method.

(iv) Obtain the output stream from the response.

(v) Get input value, `code`, from request.

(vi) If no selection is made the checkbox, display a warning message, "No selection is made" and display a hyperlink linked to `enquiry.jsp` for try again; Otherwise, Use the `ProductDB` to obtain the relevant product information and display the result as per the sample out in the **Figure Q2b**.

```
package ict.servlet;
// assume the library is properly imported
(c)(i)
public class ProductController (c)(ii){
  protected void c(iii)(HttpServletRequest request,
                        HttpServletResponse response)
            throws ServletException, IOException {
     PrintWriter out;    String action ="";
     String[]codes = null;
     // (c)(iv), (v), (vi)
  }
}
```

Q3

(a) State the methods in a servlet and briefly explain how the methods affect the servlet life cycle. [4%]

(b) Briefly describe **Four** type of scope with respect to the accessibility in JEE application. [6%]

***** END OF PAPER *****