

COMP/EECE 4882
Midterm Exam
Spring 2012

Name: Answer Key

Instructions:

- No bathroom breaks.
- No calculators.
- No cellphones—turn them off.
- No other devices.

- Closed book.
- Closed note.
- Closed neighbor.

- Verify that you have all the pages.
- Don't forget to write your name.
- Read each question carefully.
- Don't forget to answer every question.

For each of the following, circle the answer that best answers the question.

1. [3pts] Which of the following is not a problem in software engineering?
 - a. Gathering complete and valid requirements
 - b. Estimating the cost of a software project
 - c. Removing defects from software
 - d. Maintaining software over time
 - e. None of the above (i.e., all are problems)

2. [3pts] Which of the following is not an agile value?
 - a. Individuals and interactions over processes and tools
 - b. Working software over comprehensive documentation
 - c. Customer collaboration over contract negotiation
 - d. Responding to change over following a plan
 - e. None of the above (i.e., all are agile values)

3. [3pts] Which of the following problems does iterative development directly address?
 - a. Unstable requirements
 - b. Design erosion
 - c. Program comprehension
 - d. All of the above
 - e. None of the above

4. [3pts] Which of the following is not true about GET and POST?
 - a. GET is generally idempotent, and POST is generally not
 - b. POST has a data payload, and GET does not
 - c. Both GET and POST requests have request headers
 - d. GET is an HTTP method, and POST is an HTTPS method
 - e. None of the above

5. [3pts] Which of the following correctly defines what it means to be an idempotent operation?
 - a. Has side effects
 - b. Has one parameter
 - c. Can only be applied once per session
 - d. Can be applied multiple times without changing the result beyond the initial application
 - e. None of the above

6. [12pts] We learned in class that three types of information are generally communicated in the body of a user story (recall the US template). For the following user story, tell whether each type of information is present, and if so, tell what the information is. If a piece of information is missing, describe how you would reword the example to include that information (include the exact words that you would use).

US SearchForClients

As an accountant, I want to search for my clients by their first and last names.

(1) Who: An accountant

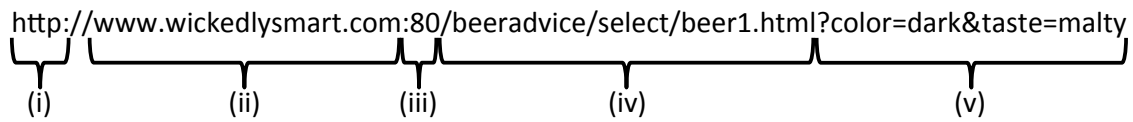
(2) What: Search for his/her clients by their first and last names

(3) Why: Missing!

To add the "Why" information, you might add the following to the end of the US:

... because I have too many clients manually search for their information, and I generally know their first and last names.

7. [5pts] For the following URL, match the part of the URL with the name for that part.



- | | | |
|------|------------------------------|-------------------------|
| i. | <u> b </u> | a. Resource (with path) |
| ii. | <u> e </u> | b. Protocol |
| iii. | <u> c </u> | c. Port |
| iv. | <u> a </u> | d. Parameters |
| v. | <u> d </u> | e. Server |

8. [3pts] Which SVN client operation creates a working copy?

- a. checkout
- b. commit
- c. update
- d. revert
- e. None of the above

9. [3pts] Which SVN client operation may report a conflict and then do nothing?

- a. checkout
- b. commit
- c. update
- d. revert
- e. None of the above

10. [3pts] Which SVN client operation should you run as the first step toward resolving a conflict, assuming your goal is to do a merge?

- a. checkout
- b. commit
- c. update
- d. revert
- e. None of the above

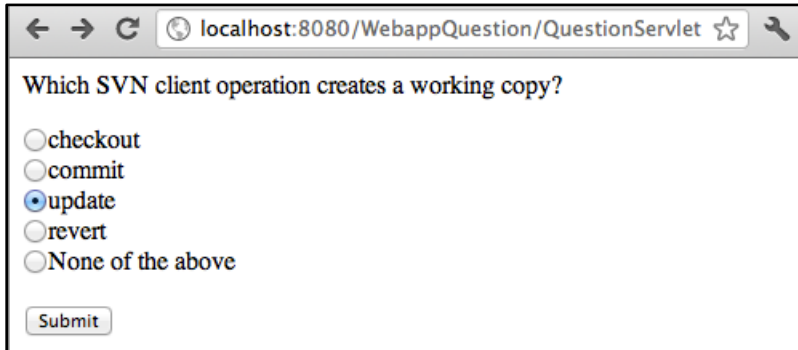
11. [6pts] For each of the following artifacts, tell which role in an MVC architecture the artifact would typically play. Give the full name of the role. (Giving only the letter M, V, or C will earn you only partial credit.)

a. A JSP View

b. A plain old Java class Model

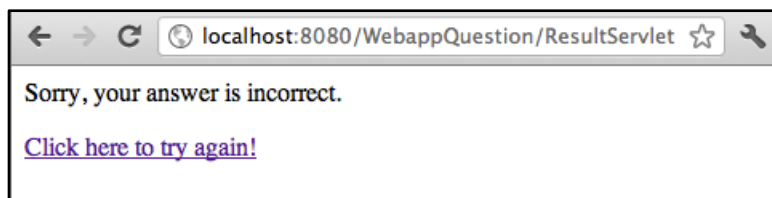
c. A servlet class Controller

12. [15pts] For this problem, you must create a servlet for a quizzing web app. The app already has a QuestionServlet servlet that responds to HTTP GET requests with a trivia question that looks like this:



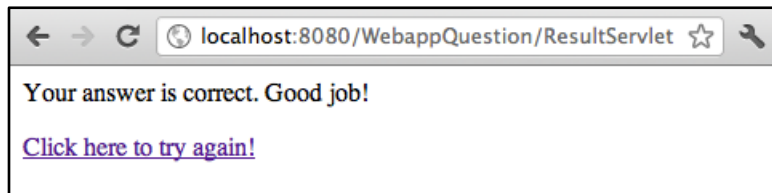
A screenshot of a web browser window. The address bar shows "localhost:8080/WebappQuestion/QuestionServlet". The page content is a trivia question: "Which SVN client operation creates a working copy?". Below the question are five radio button options: "checkout", "commit", "update" (which is selected), "revert", and "None of the above". At the bottom left of the form is a "Submit" button.

Your job is to create a ResultServlet servlet that receives the user's answer when he/she presses the submit button, and that responds with a message like this:



A screenshot of a web browser window. The address bar shows "localhost:8080/WebappQuestion/ResultServlet". The page content displays the message "Sorry, your answer is incorrect." followed by a blue underlined link that says "Click here to try again!".

or like this:



A screenshot of a web browser window. The address bar shows "localhost:8080/WebappQuestion/ResultServlet". The page content displays the message "Your answer is correct. Good job!" followed by a blue underlined link that says "Click here to try again!".

Relevant excerpts from the Java API and the HTML that QuestionServlet generated for the above trivia question is given on page 11. Your ResultServlet must use the QuestionManager class (described on page 12) to retrieve the correct answers to questions.

You need not include `import` statements or the `web.xml` file in your solution. Just write the Java code for the ResultServlet class. Also, I do not expect you to use JSPs in your solution.

Write your answer on the next page.

Write your answer to Question 12 on this page.

```
public class ResultServlet extends HttpServlet {

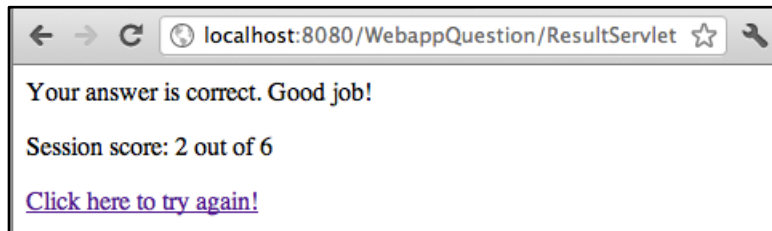
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException
    {
        Enumeration<String> params = request.getParameterNames();
        String questId = params.nextElement();
        String userAns = request.getParameter(questId);
        String correctAns = QuestionManager.getInstance().getAnswerId(questId);
        PrintWriter out = response.getWriter();

        out.println("<html>");
        out.println("<body>");
        out.println("<p>");

        if (userAns.equals(correctAns)) {
            out.println("Your answer is correct. Good job!");
        } else {
            out.println("Sorry, your answer is incorrect.");
        }

        out.println("</p>");
        out.println("<p>");
        out.println("<a href=\"QuestionServlet\">Click here to try again!</a>");
        out.println("</p>");
        out.println("</body>");
        out.println("</html>");
    }
}
```

13. [15pts] For this problem, you must enhance your previous servlet by making it keep track of the user's score for the current session:



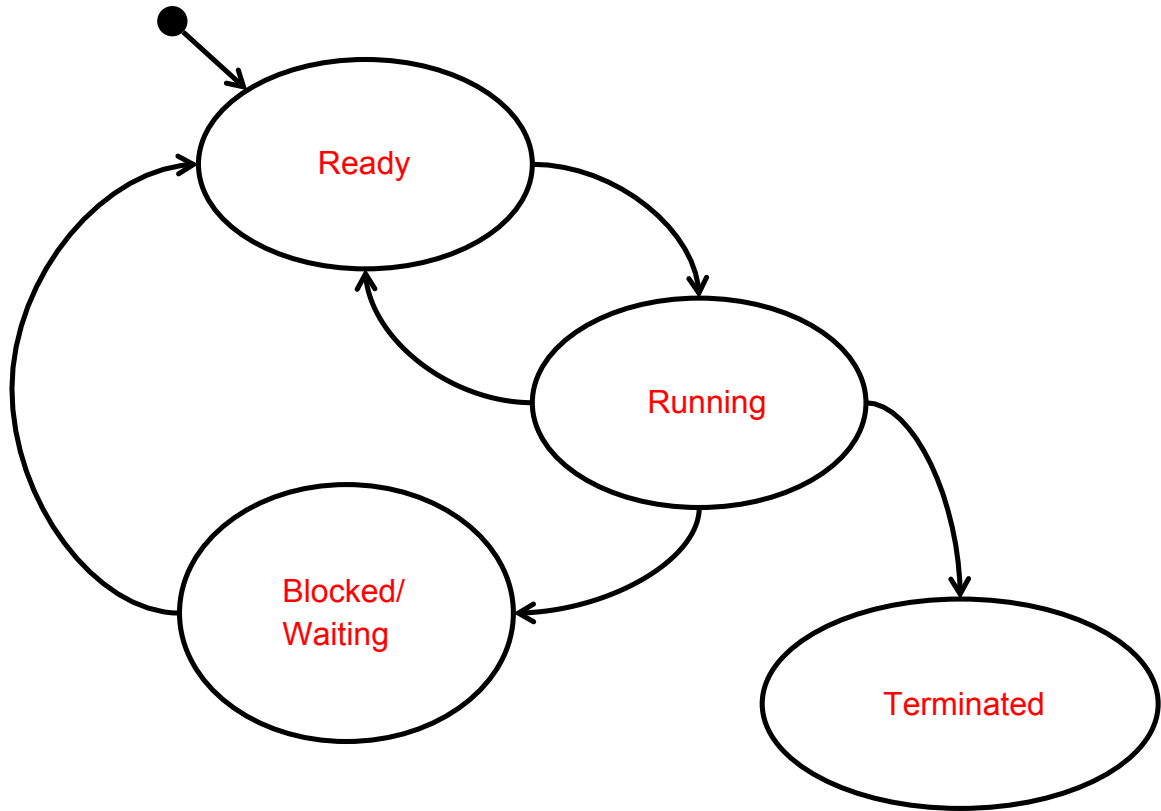
You need not rewrite all the code from your previous solution. Represent code that has not changed with ellipses (...). As before, I do not expect you to use JSPs.

```
public class ResultServlet extends HttpServlet {

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException
    {
        ...
        HttpSession session = request.getSession();

        if (session.isNew()) {
            session.setAttribute("score", new Integer(0));
            session.setAttribute("attempts", new Integer(1));
        } else {
            session.setAttribute(
                "attempts",
                new Integer(((Integer)session.getAttribute("attempts")).intValue()+1)
            );
        }
        ...
        if (userAns.equals(correctAns)) {
            ...
            session.setAttribute(
                "score",
                new Integer(((Integer)session.getAttribute("score")).intValue()+1)
            );
        } else {
            ...
        }
        ...
        out.println("<p>");
        out.println("Session score: " +
            ((Integer)session.getAttribute("score")).toString() +
            " out of " +
            ((Integer)session.getAttribute("attempts")).toString());
        out.println("</p>");
        ...
    }
}
```


14. [8pts] Complete this diagram of thread states in Java by labeling the states.



15. [15pts] List all the different output values that this program might produce.

```
public class MyThread implements Runnable {  
  
    public void run() {  
        MyMainThread.x += MyMainThread.y; // A  
        MyMainThread.y = 5; // B  
    }  
}  
  
public class MyMainThread {  
    static public int x = 1;  
    static public int y = 1;  
  
    public static void main(String[] args)  
        throws InterruptedException  
    {  
        Thread myT = new Thread(new MyThread());  
        myT.start();  
        x = 9; // C  
        y += x; // D  
        myT.join();  
        System.out.println("x = " + x);  
        System.out.println("y = " + y);  
    }  
}
```

ABCD: x = 9, y = 14
ACBD: x = 9, y = 14
ACDB: x = 9, y = 5
CABD: x = 10, y = 15
CADB: x = 10, y = 5
CDAB: x = 19, y = 5

Relevant Excerpts from the Java API

- Class HttpServlet
 - protected void service(HttpServletRequest req, HttpServletResponse resp)
 - protected void doGet(HttpServletRequest req, HttpServletResponse resp)
 - protected void doPost(HttpServletRequest req, HttpServletResponse resp)
- Interface HttpServletRequest
 - HttpSession getSession()
 - public Enumeration<String> getParameterNames()
 - public String getParameter(String name)
- Interface HttpServletResponse
 - public PrintWriter getWriter()
- Interface HttpSession
 - public boolean isNew()
 - public Object getAttribute(String name)
 - public void setAttribute(String name, Object value)
- Interface Enumeration<E>
 - boolean hasMoreElements()
 - E nextElement()

Example of HTML Generated by QuestionServlet

```
<html>
<body>
<p>
Which SVN client operation creates a working copy?
</p>
<form method="POST" action="ResultServlet">
<input type="radio" name="q1748" value="a1">checkout<br>
<input type="radio" name="q1748" value="a4">commit<br>
<input type="radio" name="q1748" value="a2">update<br>
<input type="radio" name="q1748" value="a3">revert<br>
<input type="radio" name="q1748" value="a5">None of the above
<p>
<input type="submit" value="Submit">
</p>
</form>
</body>
</html>
```

q1748 is the ID used by QuestionManager for the question. a1–a5 are IDs that QuestionMaker uses for each answer.

Class QuestionManager

```
public class QuestionManager {
    private static QuestionManager instance = null;

    public static QuestionManager getInstance() {
        if (instance == null) {
            instance = new QuestionManager();
        }
        return instance;
    }

    private QuestionManager() { ... }

    /**
     * @return The ID of the correct answer to questionId.
     */
    public String getAnswerId(String questionId) { ... }

    ...
}
```

QuestionManager follows the singleton pattern, which ensures that there is only one instance of the class in the system. To get the instance of QuestionManager, you must call its static getInstance() method (as opposed to using new).