Servlet Assignment

**Purpose : To practice using Java and servlets.**

Problem : You have been hired to write a web version of "Who Wants to Be a Millionaire". If you have not seen this show, your first assignment is to watch it so you understand this project. The basic idea is that a player is asked questions and if he or she guesses the correct answer, the player gets to go on to the next question. Otherwise, the player is finished. Here are the rules in detail:

(1) A player must "log on" to the system. When the user goes to the Millionaire URL, an HTML page is loaded which asks for the players name, city, and social security number. The page also contains submit and a clear buttons. When the submit button is selected, the information in the form is sent to the server where a servlet checks the social security number against a file of previous players. If the person has played before, they cannot play again. So, the HTML page return to the player is either :

(a) You have already played - sorry you cannot play again, or

(b) Welcome to Who Wants to Be a Millionaire and the first question. The first

question is worth $100. For example :

The pyramids are in

1. Egypt 2. Morocco 3. Canada 4. South Dakota

(2) The format of the question pages (such as the one described in (b) above) is :

Who Wants to Be a Millionaire

Question # for $xxx Current Winnings $xxx

Text of the question

1. answer 1 2. answer 2 3. answer 3 4. answer 4

Enter your answer here \_\_\_\_\_\_\_\_ (text box)

[**Better idea from class discussion: Use radio button for answer.**]

[Submit] [Clear]

(3) User will select an answer and click the submit button. The information in the form will be sent to the server. As a result of the selection, the following can happen :

Question 1 ($100)

right - return a page with question 2 for $500 and Current Winnings of $100

wrong - return a page saying sorry and indicate that winnings are $0

Question 2 ($500)

right - return a page with question 3 for $1000 and Current Winnings of $500

wrong - return a page saying sorry and indicate that winnings are $0

Question 3 ($1,000)

right - return a page with question 4 for $5000 and Current Winnings of $1,000

wrong - return a page saying sorry and indicate that winnings are $0

Question 4 ($5,000)

right - return a page with question 5 for $10,000 and Current Winnings of $5,000

wrong - return a page saying sorry and indicate that winnings are $1,000

Question 5 ($10,000)

right - return a page with question 6 for $50,000 and Current Winnings of $10,000

wrong - return a page saying sorry and indicate that winnings are $1,000

Question 6 ($50,000)

right - return a page with question 7 for $100,000 and Current Winnings of

$50,000

wrong - return a page saying sorry and indicate that winnings are $10,000

Question 7 ($100,000)

right - return a page with question 8 for $500,000 and Current Winnings of

$100,000

wrong - return a page saying sorry and indicate that winnings are $10,000

Question 8 ($500,000)

right - return a page with question 9 for $1,000,000 and Current Winnings of

$500,000

wrong - return a page saying sorry and indicate that winnings are $100,000

Question 9 ($1,000,000)

right - return a jazzy page saying Congratulations, you Won $1,000,000.

wrong - return a page saying sorry and indicate that winnings are $100,000

(4) You will need eleven files on the server : one for each question, the players file, and the winners file. Call the question files ques1.txt ... ques9.txt, call the players file players.txt and call the winners file winners.txt.

The format of the question files is :

Question (one line long)

answer 1 (each answer is one line long)

answer 2

answer 3

answer 4

correct answer

The players file format will be :

SSN

Name

City

SSN

Name

City

etc.

The format for the winners file is :

*Name* of *City* won $*xxx*.

(5) More on the files.

(a) The question files will have one question each. **Extra credit opportunity :** for 10 points of extra credit, have the program handle multiple questions on each file. Randomly select one question from each file. [**This is required and not for extra credit for two-person teams in CSC 499.**]

(b) The winner file should only contain the names of people who won something. No $0 amounts are posted. The winner file is updated after the game ends for a user.

(c) The players file is updated at the end of the game also.

Things to know :

* The six machines at the back of the McLaury lab are running a Java server that handles servlets. These machines can be used for this assignment.
* The servlets are not reloaded! You have to close and reopen when you make changes (or rename the file).
* It is handy to keep 2 DOS shells open - one for running the server (type startserver at the command prompt) and the second one for compiling. startserver will spawn another window that you can't type into. To stop the server, type stopserver in the window you have reserved for starting and stopping the server.
* In users\alogar you will find the HTTPPost Servlet example we did in class. You should make sure you can run this before you begin writing your own code. This code is also in the Deitel Java book.
* Note that in the lab the localhost setting is different. The code in users\alogar reflects this critical change.
* If you are running at home, download JSWDK from javasoft.com. Note that you may need to increase your environment space. Be sure to put servlet.jar in the right place, that is, in the jre\lib\ext directory under your jdk directory. Also, set your path.