Palindrome Practical Test

The aim of this project is to create a simple **Palindrome** Web Game (definition below). The idea of the game is to submit a word or string, and gain a certain amount of points if the sentence is a palindrome. Here is what can happen in the game:

Palindrome Definition:
A word, line, verse, number, sentence, etc, reading the same backward as forward. E.g.
Madam, I'm Adam

Poor Dan is in a droop

Do geese see God?

The player will have the possibility to submit a sentence as well as his/her name to the server.

- If the sentence is a palindrome then the user's current score must be increased by half of the size of the palindrome size.
- If the player's name is already registered in the game (if his name already exists), then add the score to the player. Else create a new record for the user. Note - The details are held in memory for a session, and do not need to be persisted to a file or database.

The player also has the possibility of getting into the "Hall of Fame", which is displayed on a web page

- This lists the 5 best players, and ranks them by score

If the server is reset, the list of user-score must be empty.

The app must be thread safe, as multiple users may be accessing it at once.

Important points

This test should concentrate on the back end server logic. Don't worry about capturing user input, integrating servlet or front end (HTML/JSP) code unless you have time left.

Please make sure the following implementation is done:
Palindrome check (Please do not use recursion!)

Storing multiple player details in memory, with the ability to rank by score
Hall of fame logic
Thread safety

Interview Theory Test

Question 1

Add code that outputs 'true' if dog1 and dog2 contain the same characters in the same order.

```
    public void testValues1(){
            String dogValue = "dog";
            String dog1 = "I am a dog";
            String dog2 = "I am a "+dogValue;
      }
```

Does Java do anything special with String constants in memory?

Question 2

- Implement a Car subclass of Vehicle
- Override Vehicle's brake() method in Car, so that it outputs "Car brake", and then makes a call to the Vehicle version of brake()
- Overload Vehicle's brake() method in Car, so that it has a parameter (String volume)
- Create a method to test the code, and do the following:
- Create a Vehicle reference that is a new Car object
- Invoke the brake() method in Car.
- Invoke the overloaded version of Car brake(String volume) without creating a new Car object

```
class Vehicle {

      public boolean canDrive = false;

      protected void beep() {
            System.out.println("Vehicle beep");
      };

      protected void brake() {
            System.out.println("Vehicle brake");
      };
}
```

Question 3

- In the example below, the output is 7. Make a change to class B (**Without adjusting the main method!**), so that the program outputs 6. Note - the value '5' is added twice
- Explain your implementation.

```
import java.util.Set;
import java.util.HashSet;

public class B
{
      private final int b;

      private B (int b)
      {
            this.b = b;
      }

    public static void
      main (String[] argv)
      {
            Set<B> set = new HashSet<B> ();

            set.add (new B (1));
            set.add (new B (2));
            set.add (new B (3));
            set.add (new B (4));
            set.add (new B (5));
            set.add (new B (5));
            set.add (new B (6));

            System.out.println (set.size ());
      }
}
```

Question 4

For user 1234, find the "detail" corresponding to the highest value of "time".

```
CREATE TABLE events (
        user INT UNSIGNED NOT NULL,
        time BIGINT UNSIGNED NOT NULL,
        detail VARCHAR(255) NOT NULL,
        PRIMARY KEY (user, time)
);
```

Question 5

- The following method does not compile. Can you suggest why?
- Make changes to the line '`x = x + i;`' without changing the order, to make the method compile.
- The value output is '-7616'. Explain why it is negative. Also, explain how Java sets an integer to negative or positive.

```
public void outputValue(){
        short x = 0;
        int i = 123456;

        x = x + i;
        System.out.println(x);
  }
```