BCDE223 SYNTAX TEST #4A

Student Name _____ Marks out of 10 _____

Markers _____

Closed book, closed directory, no notes, individual test.

Create a class called NumberHider to play a number guessing game.

- If the class is created with no parameters it should generate a random number between 0 and 99.

  Hint: To produce an int between 0 and 99, use:

  ```
  int secretNumber = (int)(Math.random()*100);
  ```
- If the class is created with an int parameter, that should be the secret number which needs to be guessed.
- The player inputs his/her guess, and the program shall response with "Try higher", "Try lower" or "You got it in n trials"

❒Two constructors provided
❒Protected attributes
❒testLowGuess passes
❒testHigh Guess passes
❒testCorrectGuess passes
❒testGuessCounting passes
❒Appropriate use of at least 1 build-in type
❒Correct use of the most appropriate Java if statements
❒Correct use of ++
❒Correct seeding of a random number

NOTE:

Do NOT create a ModelView, a ConsoleView, a Main controller or an ExerciseController

INSTEAD you MUST use the UnitTest code provided over the page

```java
import static org.junit.jupiter.api.Assertions.assertEquals;
import org.junit.jupiter.api.Test;

public class NumberHiderTest {

    private NumberHider aNumberHider;

    @Test
    public final void testLowGuess() {
        aNumberHider = new NumberHider(42);
        String expected = "Try higher";
        String actual = aNumberHider.makeGuess(1);
        String errorMessage = "Expected " + expected + " but got " + actual;
        assertEquals(expected, actual, errorMessage );
    }

    @Test
    public final void testHighGuess() {
        aNumberHider = new NumberHider(42);
        String expected = "Try lower";
        String actual = aNumberHider.makeGuess(43);
        String errorMessage = "Expected " + expected + " but got " + actual;
        assertEquals(expected, actual, errorMessage );
    }

    @Test
    public final void testCorrectGuess() {
        aNumberHider = new NumberHider(42);
        String expected = "You got it in 1 trials!";
        String actual = aNumberHider.makeGuess(42);
        String errorMessage = "Expected " + expected + " but got " + actual;
        assertEquals(expected, actual, errorMessage );
    }

    @Test
    public final void testGuessCounting() {
        aNumberHider = new NumberHider(42);
        String expected = "You got it in 3 trials!";
        aNumberHider.makeGuess(40);
        aNumberHider.makeGuess(41);
        String actual = aNumberHider.makeGuess(42);
        String errorMessage = "Expected " + expected + " but got " + actual;
        assertEquals(expected, actual, errorMessage );
    }

}
```