# Assignment 2 - Hangman

This assignment will test your ability to use loops and work with strings in Java.

## 1 The Assignment

Recreate the Hangman game in Java, using the following rules (see also the expected input/output below):

- Player one enters a word that player two will have to try and guess.
- The program will then print out a blanked out string of the same length as the word.
- Player two then guesses a letter.
- If the letter is in the word, it gets filled in, and if not, player two looses a guess. Player two has eight guesses in total.
- If player two tries to guess a letter that they have already guessed, they do not lose a guess, and a message is printed to let them know they have already guessed that letter.
- Player two continues to guess until they have filled in all the letters, or run out of guesses.
- If player two fills in all the letters before they have used up all of their guesses, they are the winner.
- If player two runs out of guesses before filling in all of the letters, player one is the winner.

### 1.1 Instructions

Write the class `Hangman` in the file `Hangman.java`.

## 1.2 Tips

- Remember that your class name needs to be exactly the same as the name of the file in which it is defined.
- Every string that is printed must end with a newline character.
- The `System.out.println` method automatically appends a newline character to the string it prints.
- The `.charAt()` and `.indexOf()` methods of Java's `String` class will be very useful in this assignment.
- The `String.indexOf(char)` method returns `-1` if the designated char is not found in a string.
- Depending on the font you are using, it might seem like you need to add spaces between underscore characters. DO NOT add spaces between the letters, rather just change the IDE's font to Source Code Pro.
- Note that there is a difference in the sample I/O between what the program is meant to print out depending on whether player two wins or loses.

## 1.3 Expected Input/Output

### 1.3.1 Example 1

```
OUT > Player One, enter a word:
IN  > random
OUT > ______
OUT > Player Two, you have 8 guesses left. Enter a guess:
IN  > a
OUT > _a____
OUT > Player Two, you have 8 guesses left. Enter a guess:
IN  > o
OUT > _a__o_
OUT > Player Two, you have 8 guesses left. Enter a guess:
IN  > r
OUT > ra__o_
OUT > Player Two, you have 8 guesses left. Enter a guess:
IN  > n
OUT > ran_o_
```

```
OUT > Player Two, you have 8 guesses left. Enter a guess:
IN  > r
OUT > You have already guessed 'r'.
OUT > ran_o_
OUT > Player Two, you have 8 guesses left. Enter a guess:
IN  > d
OUT > rando_
OUT > Player Two, you have 8 guesses left. Enter a guess:
IN  > m
OUT > random
OUT > Game over. Player Two wins!
```

### 1.3.2 Example 2

```
OUT > Player One, enter a word:
IN  > rythm
OUT > _____
OUT > Player Two, you have 8 guesses left. Enter a guess:
IN  > a
OUT > _____
OUT > Player Two, you have 7 guesses left. Enter a guess:
IN > o
OUT > _____
OUT > Player Two, you have 6 guesses left. Enter a guess:
IN  > e
OUT > _____
OUT > Player Two, you have 5 guesses left. Enter a guess:
IN  > i
OUT > _____
OUT > Player Two, you have 4 guesses left. Enter a guess:
IN  > u
OUT > _____
OUT > Player Two, you have 3 guesses left. Enter a guess:
IN  > p
OUT > _____
OUT > Player Two, you have 2 guesses left. Enter a guess:
IN  > s
OUT > _____
OUT > Player Two, you have 1 guesses left. Enter a guess:
IN  > x
```

```
OUT > Game over. Player One wins! The word was: rythm
```

### 1.3.3 Example 3

```
OUT > Player One, enter a word:
IN  > lol
OUT > ___
OUT > Player Two, you have 8 guesses left. Enter a guess:
IN  > l
OUT > l_l
OUT > Player Two, you have 8 guesses left. Enter a guess:
IN  > l
OUT > You have already guessed 'l'.
OUT > l_l
OUT > Player Two, you have 8 guesses left. Enter a guess:
IN  > o
OUT > lol
OUT > Game over. Player Two wins!
```

## 2 Submitting to the Autograder

1. Complete your assignment, making sure your program's output matches the expected output stipulated by the assignment brief.
2. Make sure that your program compiles and runs without any errors.
3. Create a `zip` file containing the `Hangman.java` file.
4. Upload the `zip` file to Athena.

## 3 Bonus Marks

Allow player two to guess an entire word, rather than just one letter at a time. For every letter in the guessed word that does not occur in the target word, and has not already been guessed, player one should loose a guess. Every letter in the guessed word that is in the target word and has not already been guessed should be filled in. If player two only has one guess left, and there are multiple letters still left, they should be able to win by guessing a whole word. If a whole word is entered and some of the letters have already been guessed, you DO NOT need to print out the message informing the player that those letters have been guessed.

### 3.1 Expected Input/Output

#### 3.1.1 Example 1

```
OUT > Player One, enter a word:
IN  > rythm
OUT > _____
OUT > Player Two, you have 8 guesses left. Enter a guess:
IN  > socks
OUT > _____
OUT > Player Two, you have 4 guesses left. Enter a guess:
IN  > rythm
OUT > Game over. Player Two wins!
```

#### 3.1.2 Example 2

```
Player One, enter a word:
OUT > golden
IN  > ______
OUT > Player Two, you have 8 guesses left. Enter a guess:
IN  > a
OUT > ______
OUT > Player Two, you have 7 guesses left. Enter a guess:
IN  > i
OUT > ______
OUT > Player Two, you have 6 guesses left. Enter a guess:
IN  > e
OUT > ____e_
OUT > Player Two, you have 6 guesses left. Enter a guess:
IN  > o
OUT > _o__e_
OUT > Player Two, you have 6 guesses left. Enter a guess:
IN  > u
OUT > _o__e_
OUT > Player Two, you have 5 guesses left. Enter a guess:
IN  > t
OUT > _o__e_
OUT > Player Two, you have 4 guesses left. Enter a guess:
IN  > s
OUT > _o__e_
OUT > Player Two, you have 3 guesses left. Enter a guess:
```

```
IN  > o
OUT > You have already guessed 'o'.
OUT > _o__e_
OUT > Player Two, you have 3 guesses left. Enter a guess:
IN  > l
OUT > _ol_e_
OUT > Player Two, you have 3 guesses left. Enter a guess:
IN  > d
OUT > _olde_
OUT > Player Two, you have 3 guesses left. Enter a guess:
IN  > b
OUT > _olde_
OUT > Player Two, you have 2 guesses left. Enter a guess:
IN  > f
OUT > _olde_
OUT > Player Two, you have 1 guesses left. Enter a guess:
IN  > golden
OUT > Game over. Player Two wins!
```