Project 1: Hangman (A Word Guessing Game)

Introduction

In this project, you are going to implement the word guessing game, Hangman. Your program is going to get a random word from an enumerated list (instructions below) and then let the user guess letters until they get the word correct, run out of guesses, or solve the word. A unique characteristic of Hangman is that the user must specify a subset of the available spaces to check. So, for each guess, the user provides the letter to guess along with the spaces he/she wants to check.

The game has three levels of difficulty. The difficulty level determines the number of spaces allowed to check on each guess along with the total number of guesses the user can make. At the easy difficulty level, the user is allowed 15 guesses and must specify 4 spaces to check per guess. Users playing at the intermediate difficulty level get 12 guesses and must specify 3 spaces. When the game is on the hard difficulty level, the user gets 10 guesses and must specify 2 spaces at a time. An incorrect guess (or solve attempt) causes the program to decrement the number of remaining guesses, but a correct guess does not affect the number of remaining guesses. Also, invalid input does not affect the number of remaining guesses.

Write a Java program in a file called **Hangman.java** that allows the user to play Hangman as many times as they wish as long as the user doesn't exceed 20 games (the maximum number of games the user can play – that's how many items are in the word bank). When the program begins, it will ask for the difficulty level from the user. The program should read the user's input as a String, take the first letter from the string, and make sure that it is equal to 'e', 'i', or 'h'. If the input is invalid, then a message is printed, and the user is re-prompted until the input is valid. The program then asks the user for their guess. At any point, the user can choose to "solve" the word instead of entering another level. If the user enters "solve" with any capitalization, a message should be printed allowing the user to solve the game. If this answer is incorrect, the number of guesses should be decremented and the user should be re-prompted to enter a letter. If the user correctly solves the word, the game should end. After the game ends, the program will print the result and ask the user if they want to play again. Many other scenarios are given in the examples below. Your program should behave the same way for all test cases given. **You**

should also come up with additional test cases in order to thoroughly test your application.

Sample Output

Each example has the output of a single run of a correctly working program. In these examples, testingMode is on, which means that you can see the secret word before guessing (see below for more details). This is very handy when debugging your code. The user's input in the examples is marked in green.

Example 1 (User wins by guessing letters):

```
Enter your difficulty: Easy (e), Intermediate (i), or Hard (h)
eeeeeeeeee
The secret word is: identifier
The word is: ----------
Please enter the letter you want to guess: d
Please enter the spaces you want to check (separated by spaces):
0 1 2 5
Your guess is in the word!
The updated word is: -d--------
Guesses Remaining: 15
Please enter the letter you want to guess: i
Please enter the spaces you want to check (separated by spaces):
0 1 5 7
Your guess is in the word!
The updated word is: id---i-i--
Guesses Remaining: 15
Please enter the letter you want to guess: e
Please enter the spaces you want to check (separated by spaces):
2 3 8 9
Your guess is in the word!
The updated word is: ide--i-ie-
Guesses Remaining: 15
Please enter the letter you want to guess: n
Please enter the spaces you want to check (separated by spaces):
3 4 5 6
Your guess is in the word!
The updated word is: iden-i-ie-
Guesses Remaining: 15
Please enter the letter you want to guess: t
Please enter the spaces you want to check (separated by spaces):
3 4 5 6
Your guess is in the word!
The updated word is: identi-ie-
Guesses Remaining: 15
Please enter the letter you want to guess: f
Please enter the spaces you want to check (separated by spaces):
4 5 6 7
```

```
Your guess is in the word!
The updated word is: identifie-
Guesses Remaining: 15
Please enter the letter you want to guess: r
Please enter the spaces you want to check (separated by spaces):
6 7 8 9
Your guess is in the word!
The updated word is: identifier
Guesses Remaining: 15
You have guessed the word! Congratulations
Would you like to play again? Yes(y) or No(n)
y
Enter your difficulty: Easy (e), Intermediate (i), or Hard (h)
…… // execution continues for the next round
```

Example 2 (Checking invalid input)

```
Enter your difficulty: Easy (e), Intermediate (i), or Hard (h)
f
Invalid difficulty.  Try Again...
Enter your difficulty: Easy (e), Intermediate (i), or Hard (h)
g
Invalid difficulty.  Try Again...
Enter your difficulty: Easy (e), Intermediate (i), or Hard (h)
e
The secret word is: identifier
The word is: --------
Please enter the letter you want to guess: 7
Your input is not valid.  Try again.
Guesses Remaining: 15
Please enter the letter you want to guess: a
Please enter the spaces you want to check (separated by spaces):
1 2
Your input is not valid.  Try again.
Guesses Remaining: 15
Please enter the letter you want to guess: a
Please enter the spaces you want to check (separated by spaces):
1 2 3
Your input is not valid.  Try again.
Guesses Remaining: 15
Please enter the letter you want to guess: a
Please enter the spaces you want to check (separated by spaces):
1 2 3 4 5
Your input is not valid.  Try again.
Guesses Remaining: 15
Please enter the letter you want to guess: b
Please enter the spaces you want to check (separated by spaces):
1 5 9 12
Your input is not valid.  Try again.
```

```
Guesses Remaining: 15
…… //Execution continues from this point.
```

Example 3 (If two letters are entered as the guess, take the first one)

```
Enter your difficulty: Easy (e), Intermediate (i), or Hard (h)
e
The secret word is: identifier
The word is: ----------
Please enter the letter you want to guess: id
Please enter the spaces you want to check (separated by spaces):
0 1 2 3
Your guess is in the word!
The updated word is: i---------
Guesses Remaining: 15
Please enter the letter you want to guess://Execution continues beyond this point.
```

Example 4 (If user makes an identical correct guess or a guess containing an already uncovered space, don't decrement guesses remaining)

```
Enter your difficulty: Easy (e), Intermediate (i), or Hard (h)
e
The secret word is: identifier
The word is: ----------
Please enter the letter you want to guess: i
Please enter the spaces you want to check (separated by spaces):
0 0 0 0
Your guess is in the word!
The updated word is: i---------
Guesses Remaining: 15
Please enter the letter you want to guess: i
Please enter the spaces you want to check (separated by spaces):
0 1 2 3
Your guess is in the word!
The updated word is: i---------
Guesses Remaining: 15
Please enter the letter you want to guess: i
Please enter the spaces you want to check (separated by spaces):
1 2 3 4
Your letter was not found in the spaces you provided.
Guesses Remaining: 14
Please enter the letter you want to guess: i
Please enter the spaces you want to check (separated by spaces):
3 4 5 6
Your guess is in the word!
The updated word is: i----i----
Guesses Remaining: 14
Please enter the letter you want to guess: //Execution continues from this point
```

Example 5 (Max guesses is exceeded, resulting in a player loss)

```
Enter your difficulty: Easy (e), Intermediate (i), or Hard (h)
h
The secret word is: identifier
The word is: ----------
Please enter the letter you want to guess: z
Please enter the spaces you want to check (separated by spaces):
0 1
Your letter was not found in the spaces you provided.
Guesses Remaining: 9
Please enter the letter you want to guess: z
Please enter the spaces you want to check (separated by spaces):
1 2
Your letter was not found in the spaces you provided.
Guesses Remaining: 8
Please enter the letter you want to guess: z
Please enter the spaces you want to check (separated by spaces):
4 5
Your letter was not found in the spaces you provided.
Guesses Remaining: 7
Please enter the letter you want to guess: q
Please enter the spaces you want to check (separated by spaces):
0 3
Your letter was not found in the spaces you provided.
Guesses Remaining: 6
Please enter the letter you want to guess: r
Please enter the spaces you want to check (separated by spaces):
0 1
Your letter was not found in the spaces you provided.
Guesses Remaining: 5
Please enter the letter you want to guess: z
Please enter the spaces you want to check (separated by spaces):
y
Your input is not valid. Try again.
Guesses Remaining: 5
Please enter the letter you want to guess: y
Please enter the spaces you want to check (separated by spaces):
2 3
Your letter was not found in the spaces you provided.
Guesses Remaining: 4
Please enter the letter you want to guess: h
Please enter the spaces you want to check (separated by spaces):
4 5
Your letter was not found in the spaces you provided.
Guesses Remaining: 3
Please enter the letter you want to guess: d
Please enter the spaces you want to check (separated by spaces):
7 8
Your letter was not found in the spaces you provided.
Guesses Remaining: 2
```

```
Please enter the letter you want to guess: u
Please enter the spaces you want to check (separated by spaces):
0 3
Your letter was not found in the spaces you provided.
Guesses Remaining: 1
Please enter the letter you want to guess: a
Please enter the spaces you want to check (separated by spaces):
3 4
Your letter was not found in the spaces you provided.
Guesses Remaining: 0
You have failed to guess the word... :(
Would you like to play again? Yes(y) or No(n)
n
```

Example 6 (Solving word)

```
Enter your difficulty: Easy (e), Intermediate (i), or Hard (h)
i
The secret word is: identifier
The word is: ----------
Please enter the letter you want to guess: i
Please enter the spaces you want to check (separated by spaces):
0 4 5
Your guess is in the word!
The updated word is: i----i----
Guesses Remaining: 12
Please enter the letter you want to guess: solve
Please solve the word: identical
That is not the secret word.
Guesses Remaining: 11
Please enter the letter you want to guess: d
Please enter the spaces you want to check (separated by spaces):
0 1 2
Your guess is in the word!
The updated word is: id---i----
Guesses Remaining: 11
Please enter the letter you want to guess: solve
Please solve the word: identifier
You win!
You have guessed the word! Congratulations
Would you like to play again? Yes(y) or No(n)
n
```

Requirements:

- You **must** use index 0 as the index of the first character in your word. For example, if the secret word is “hardware”, the letter ‘h’ is at index 0. So, if the user specifies space 0, you should uncover the ‘h’. Failure to use correct indexing will have a negative

impact on your assignment grade.

- In order to facilitate the testing of your program, you **must** include a boolean variable called testingMode initialized to true in the top of your class (under the class declaration and above your main method) as shown below.

```
private static final boolean testingMode = true;
```

  If the value of the variable testingMode is true, your program will display the secret word the user should guess. This helps you test your program and helps us grade your program. Without this variable, we will not know what the correct answer is. See example 1 to see the output. On the other hand, if the value of the variable testingMode is set to false, the program will not show the value of the secret word. When you submit your code, make sure the variable is set to true.

- Use the RandomWord.newWord() method to generate random words, and this method should only be called once per game, and it can be called at most 20 times in a single program run (note: after 20 calls, it issues an error message and terminates). This method is provided to you. However, you must download the RandomWord.java file from the labs and projects website, and place RandomWord.java in the same source directory as Hangman.java. RandomWord.java randomly picks a word from an enumerated list and returns it when the the newWord() method is called. If you are interested in how it works, you can look at the source code. Do not modify the source code in RandomWord.java.
- The code for this assignment will require loops, decision statements, variables, etc.

Hints:

- You may want to use two Strings to store the secret word and the word that is displayed to the user. The secret word is stored internally (the user can't see it unless testingMode is set to true) and the display word is what they see on the screen. The display word will start out as a series of dashes ("-") and then be modified after a correct guess.
- You may find the following methods useful:
  - Character.isLetter(*char arg*) – returns true if the character arg is a letter
  - Character.isDigit(*char arg*) – returns true if the character arg is a number
  - Character.getNumericValue(*char arg*) – returns the numeric value of a character

Additional Requirements:

You must make your program output look **exactly** like the examples above.

Your program must use RandomWord.newWord() method in the provided RandomWord class. Failure to use this method may result in a failing grade. Also, your testingMode boolean variable must be set to true when you submit your final version.

The name of the class in your java program must be Hangman. Therefore, the java source code file must be called **Hangman.java**.

You are also required to evaluate contributions of your teammates. Suppose there are four members in your team, you need to submit a txt file (**`evaluation.txt`**) clarifying each member's contribution:

- Full name of member A: 20%
- Full name of member B: 30%
- Full name of member C: 10%
- Full name of member D: 40%

Project Submission and Grading

Team collaboration is required for project assignments. A team allows for maximum five students. **It is required that you, if you are the team leader, set up a team repository on GitLab for each project, add other team members as developers, and add both the instructor and the TA as reporters**. *Note that the team repository should be different from the repository of your lab assignments*.

When your team repository is set up, please fill this information collection query (https://goo.gl/forms/1OjPkLDWkVBlNP402) so we can match your team information with grading scheme. **Failure to do so could result in a grade of zero**.

All projects are graded out of a possible 100 points:

- Please use comments to explain your program or any sections that are possibly difficult to understand. **10 points would be deducted if few or no comments can be found in your java files**.
- **30 points would be deducted for each project if there are no commit/push in your GitLab repository by 11:59 pm, Oct. 19th**.
- **Programs that do not compile will receive a grade of zero**. You must make absolutely certain your program compiles before submitting, and you must thoroughly test your program with different inputs to verify that it is working correctly. Your program must work with aforementioned RandomWord.java file; otherwise, your project may receive a failing grade.
- All instructions must be followed in order to receive full credit. Your program will be tested thoroughly to verify if it plays a series of word guessing runs correctly and follows the directions in the above description (**especially the error handling cases**).

Submit a .zip file composed of the followings via **Canvas** after you thoroughly test your project:

- **`Hangman.java`**
- **`evaluation.txt`**
- **`Other java source code files if deemed necessary (e.g., a supporting class for Hangman.java); Please note that we will execute only Hangman.java for grading purposes.`**

The project submitted through **Canvas** will be graded according to the following guidelines.

- A score between 0 and 100 will be assigned.
- If the source file(s) are not submitted by the deadline, then a grade of 0 will be assigned.
- 10 points will be deducted for late labs per day.
    - 0-24 hours late -10
    - 24-48 hours late -20
    - >48 hours late = 0