

---

# System Requirements Specification Index

For

## Java Skills Evaluation

Version 1.0

## USE CASE DESCRIPTION

---

### Use Case 1

Write a code to convert a string into its piglatin encode.

Example:

Input: This is a string

Output: hisTa sia aa tringsa

#### Description

1. Take console input of a string, in **main ()** method, from user
2. Pass the string to the method **getPiglatin()** and write the logic in that method to convert it into piglatin.
3. Return the piglatin string from **getPiglatin()** to **main()** and display the piglatin string.
4. Template Code of the same is provided at:  
**com.iiht.evaluation.yaksha.piglatin.MyApp**

### Use Case 2

Write a code to convert a number into individual words

Example:

Input: 2435

Output: TWO FOUR THREE FIVE

#### Description

1. Take console input of a number, in **main ()** method from the user.
2. Pass the number to method **getWords()** and write the logic in that method to convert numbers into individual words and store them in a single string.
3. Return the string from **getWords()** to **main()** and display the string.
4. Template Code of the same is provided at:  
**com.iiht.evaluation.yaksha.words.MyApp**

## EXECUTION STEPS

---

1. All actions like build, compile, running application, running test cases will be through Command Terminal.
2. To open the command terminal the test takers, need to go to Application menu (Three horizontal lines at left top) -> Terminal -> New Terminal.
3. This editor Auto Saves the code.
4. If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use CTRL+Shift+B- command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.
5. These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.
6. Command to run code for use-case 1:  

```
mvn compile exec:java -Dexec.mainClass="com.iiht.evaluation.yaksha.piglating.MyApp"
```
7. Command to run code for use-case 2:  

```
mvn compile exec:java -Dexec.mainClass="com.iiht.evaluation.yaksha.words.MyApp"
```
8. You need to use CTRL+Shift+B - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.

## CONSIDERATIONS

---

1. **Mandatory:** Run the following command before the final submission

**mvn test**

(You can run this command multiple times to identify the test case status, and refactor code to make maximum test cases passed before final submission)

2. Your code will also be evaluated for code quality, naming conventions, readability etc.
3. **Make sure you do not modify existing class and method names and their signatures, else it would severely affect final result**
4. **Make sure you do not add any new class or methods, else it would severely affect final result**
5. **Make sure you do not modify any test cases, else it would severely affect final result**