Java Loop Through a HashMap - Assignment

Instructions:

You are provided with the `HashMapOperations.java` class. Your task is to implement the `main()` method using the necessary HashMap operations. Below are the specific tasks you need to complete, including exact variable names and type declarations.

Task 1: Creating a HashMap

1. **Objective**: Create a `HashMap` and add items to it.

2. **Details**:
   - You need to create a `HashMap` with `String` as the key type and `Integer` as the value type.
   - Add the following key-value pairs to the `HashMap`:
     - `"One"` → `1`
     - `"Two"` → `2`
     - `"Three"` → `3`

3. **Steps**:
   - **Create a `HashMap`** named `hashMap` with key type `String` and value type `Integer`.
   - Use the `put()` method to add the following key-value pairs to the `hashMap`:
     - `"One"` → `1`
     - `"Two"` → `2`
     - `"Three"` → `3`
   - After adding the items, print the entire `hashMap` to the console as ["HashMap after adding items: " + hashMap].

Task 2: Accessing an Item from the HashMap

1. **Objective**: Access an item from the `HashMap` using the `get()` method.

2. **Details**:
   - You need to access the value of the key `"Two"` from the `HashMap`.

3. **Steps**:
  - Use the `get()` method to retrieve the value associated with the key `"Two"`.
  - Store the retrieved value in a variable named `value` of type Integer.
  - Print the value to the console as ["Value of key 'Two': " + value].

Task 3: Removing an Item from the HashMap

1. **Objective**: Remove an item from the `HashMap` using the `remove()` method.

2. **Details**:
  - You need to remove the entry with the key `"Three"` from the `HashMap`.

3. **Steps**:
  - Use the `remove()` method to remove the entry with the key `"Three"`.
  - Print the `hashMap` after removing the item to verify the change as ["HashMap after removing item with key 'Three': " + hashMap].

Task 4: Looping Through the HashMap

1. **Objective**: Loop through the `HashMap` and print each key-value pair.

2. **Details**:
  - You need to use a `for-each` loop to iterate over the entries in the `HashMap`.

3. **Steps**:
  - Use the `entrySet()` method to get the set of entries in the `hashMap`.
  - Use a `for-each` loop to iterate through the `Map.Entry<String, Integer>` entries.
  - Print the key and value of each entry as [key + " => " + value]

Final Deliverable:

1. Implement the `main()` method in the `HashMapOperations.java` class.
2. Ensure that:
  - You **create a `HashMap`** named `hashMap` with key type `String` and value type `Integer`.
  - You **add the following key-value pairs**: `"One"` → `1`, `"Two"` → `2`, `"Three"` → `3`.
  - You use the `get()` method to **retrieve the value** associated with the key `"Two"` and print it.
  - You use the `remove()` method to **remove the entry** with the key `"Three"`.
  - You use a `for-each` loop to **iterate through** the `HashMap` and print each key-value pair.

3. Do not include any print statements outside of the specified operations.

Execution Steps to Follow:

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. **To run your project use command:**

   mvn compile exec:java -Dexec.mainClass="com.yaksha.assignment.HashMapOperations"

7. **To test your project test cases, use the command**

   mvn test

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.**