

```
// This program is copyright VUW.
// You are granted permission to use it to construct your answer to a COMP102 assignment.
// You may not distribute it in any other way without permission.

/* Code for COMP102 - 2024T3, Assignment 2
 * Name:
 * Username:
 * ID:
 */

import ecs100.*;
import java.awt.Color;

/**
 * Asks user for a key word and checks whether it meets a set of rules or not.
 */

public class KeyValidator {

    /**
     * Asks user for key word and then checks if it is a valid key word.
     */
    public void doCore(){
        UI.clearText();
        String key = UI.askString("Key:  ");
        UI.println();
        this.validateKeyCore(key);
    }

    /** CORE
     * Report "Valid" or "Invalid: ...reason...."
     */
    public void validateKeyCore(String key){
        /*# YOUR CODE HERE */

        if (key.length() < 6) {
            UI.println(key + ":key word is too short");
        }
        else if (key.length() > 12) {
            UI.println(key + ":key word is too long");
        }
        else if (key.startsWith("%") || key.startsWith("&")){
            UI.println(key + ":key word starting with '%' or '&' is invalid");
        }
        else if (key.contains("_")){
            UI.println(key + ":key word must not contain '@' anywhere");
        }
        else {
            UI.println("Your key is valid");
        }

    }

    /**
     * Asks user for key word and the name and then checks if it is a valid key word.
     */
    public void doCompletion(){
        UI.clearText();
        String key1 = UI.askString("Key:  ");
        UI.println();
        this.validKeyCompletion(key1);
    }
    public void validKeyCompletion(String key1){
```

```

    if (key1.length() < 6) {
        UI.println(key1 + ":key word is too short");
    }
    else if (key1.length() > 12) {
        UI.println(key1 + ":key word is too long");
    }
    else if (key1.startsWith("%") || key1.startsWith("&")){
        UI.println(key1 + "::key word starting with '%' or '&' is invalid");
    }
    else if (key1.contains("_")){
        UI.println(key1 + ":key word must not contain '@' anywhere");
    }
    else if (key1.equals(key1.toLowerCase())) {
        UI.println(key1 + ":key word must have Upper case characters");
    }
    else if (key1.equals(key1.toUpperCase())) {
        UI.println(key1 + ":key word must have Lower case characters");
    }

    else if ((key1.contains("%") && !key1.contains("$")) || (key1.contains("$") &&
!key1.contains("%"))) ) {
        UI.println(key1 + " should contain a % or a $ ,but not both");
    }

    else {
        UI.println("Your key is valid");
    }
}

/** COMPLETION
 * Report that the key is valid or report ALL the rules that the key failed.
 */

/**
 * Setup GUI and buttons
 */
public void setupGUI(){
    UI.initialise();
    UI.addButton("Clear", UI::clearText );
    UI.addButton("Validate Key Core", this::doCore );
    UI.addButton("Validate Key Completion", this::doCompletion );
    UI.addButton("Quit", UI::quit );
    UI.setDivider(1);      // Expand the text area
}

/**
 * Create object and call setupGUI on it
 */
public static void main(String[] args){
    KeyValidator kv = new KeyValidator();
    kv.setupGUI();
}

```

```
}  
}
```