**ITE3101 Introduction to Programming**

**Programming Assignment Due Date: 21 Nov 2016**

# Notice to Students

1. This assignment should be written by the individual student. All downloaded materials are not allowed.

2. Plagiarism will be treated seriously. All assignments that have been found involved wholly or partly in plagiarism (no matter these assignments are from the original authors or from the plagiarists) will score **ZERO** marks.

3. Your program must use Java JDK8 or above to develop.

4. Your program must be structured and well commented.

5. You will receive a key by IVE email, and you must replace the key with REPLACE\_THIS\_WITH\_YOUR\_KEY.

/\*

<key>

REPLACE\_THIS\_WITH\_YOUR\_KEY

</key>

\*/

**ZERO** mark, if you cannot put a valid key or for any duplicated keys!   
**DO NOT SHARE YOUR CODE TO ANYONE, ELSE BOTH OF YOU WILL GET ZERO MARK!**

6. Write down your test cases and the reason(s) for them. Test your program by using your test cases. (You are not required to write Java Test case!)

7. You are required to hand a zip file containing the following 2 items via the assignment link in Moodle:

a. The program code of your java file, including detailed comments

b. Input file to show your test cases

8. The weight of this assignment is 20% of the module total assessment.

9. You need to submit your code progress, but not just the final code to prove your workload, and code monitoring program will capture your progress all the time.  
(Please save or run your source code regularly!)

10. The system will detect any abnormal behavior, and students are required to explain his/her code line by line verbally to prove there is no plagiarism.

11. The system will test your code regularly, check the code similarity. Common or same answer will only receive average marks.

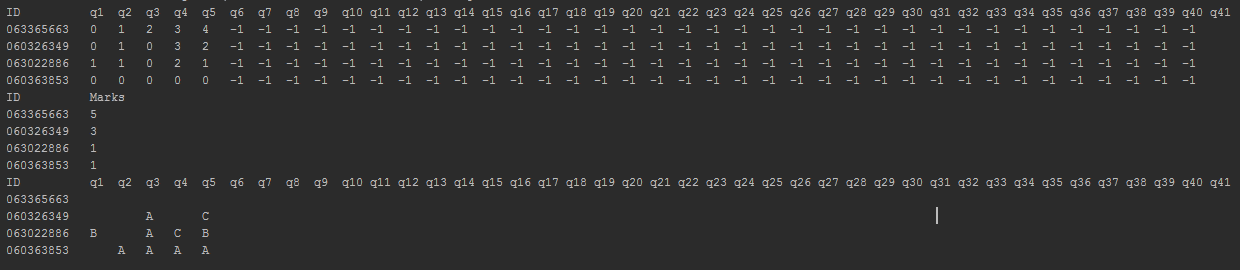
(Test documents are subject to change, and you will be reminded, if there is any update!)

Automatic Multiple Choices Marking System

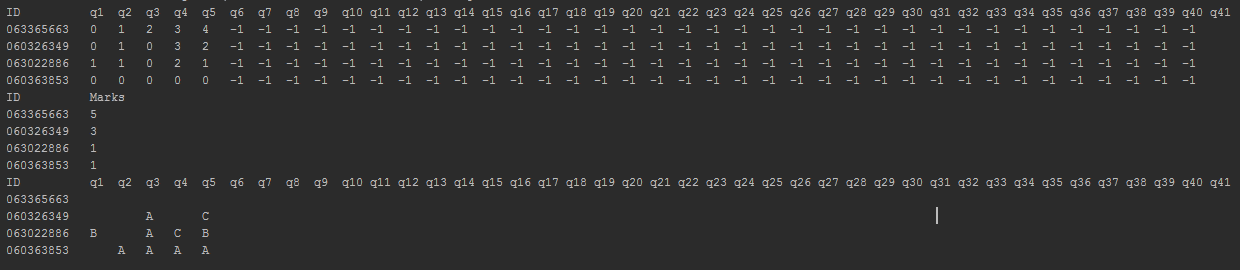
Background

This java program mainly focuses on Multiple Choices (MC) Marking and generate several reports.

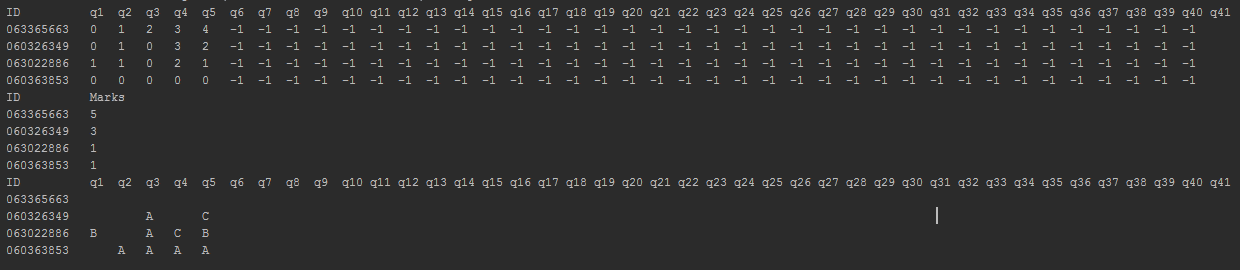
1. Test folder contains
   1. One valid format multiple choices file called Answer.xlsx, which is the standard answer for that multiple choices test.
   2. Any number of students MC answer sheets (i.e. \*.xlsx), and one file per student.
2. Number of questions is based on the filled in Question count in Answer.xlsx.
3. Expected Output 3 Reports:

Student Answer ReportIt displays all student answers - 0 = A, 1 = B, 2 = C, 3 = D, 4 = E and -1 for other invalid cases such as no answer anything or filled multiple sheet.

Student Total Mark Report



It displays the total mark for each student.

Wrong Answer ReportIt displays the wrong answer and X for the invalid cases.

**Requirements**

Given several sets of excel files, write a Java program to generate reports. Please check the sample files from Moodle.

*In order to get the mark, you have to pass the predefined test cases, and you have to ensure your code can build anytime!*

**Tasks**

Follow the calling order in “main” method, and fill in defined methods according to the following table description.

#### readFolderPath (3 marks)

public static String readFolderPath()

Prompt user to input folder that contains all excel files.  
1. Create Scanner.  
2. Display message "Please input assignment folder path:"  
3. get user input folder path and return the input value.

**Returns:**

folder path that contains all mc excels files.

#### printRow (4 marks)

public static void printRow(String[] row)

Print row array and separate with the tab.  
1. Loop through row  
2. Print each row value- if it is not null, and for null, print empty string ("").  
3. Print tab after each value.

**Parameters:**

row - it is String array.

#### print2DStringArray (3 marks)

public static void print2DStringArray(String[][] answerExcel)

Print 2D Array for display excels answer.  
1. Loop through answerExcel.  
2. Call printRow for each item.  
3. Print new line.

**Parameters:**

answerExcel - 2D String Array stores students answer.

#### getWrongAnswers (15 marks)

public static char[][] getWrongAnswers(int[] answers,

int[][] studentAnswers)

In addition to the total marks, teachers are interested to know how students are getting wrong!  
You have to use nested for loop to compare standard answer and student answer.  
Requirements:  
1. Return a 2D char array that stores the wrong answer.  
2. if student answer is wrong, then store the wrong answer.  
3. if the standard answer is equal to -1 (the question is not used!), then store ' '.  
4. if student answer is equal to -1 (student fills in something wrong i.e. filled 2 cells!), then store 'X'  
5. You must use “switch” to handle the logic!

**Parameters:**

answers - int array that represents the standard answer.

studentAnswers - 2D student answer array

**Returns:**

2D char array that shows how students are getting wrong!

#### getTotalMarks (7 marks)

public static int[] getTotalMarks(int[] answers,

int[][] studentAnswers)

Get the total mark for each student.  
1. Create array totalMarks.  
2. Nested for loop to calculate total mark, and you have to take care of -1, and correct answer.  
3. Return totalMarks

**Parameters:**

answers - correct answer array

studentAnswers - 2D student answer array

**Returns:**

total mark array

#### printTotalMarkReport (3 marks)

public static void printTotalMarkReport(String[] studentIds,

int[] totalMarks)

Print Total mark report.  
1. First line is "ID\t\t\tMarks"  
2. Loop through both studentIds and totalMarks. i.e. ID mark

**Parameters:**

studentIds - String Array of student id.

totalMarks - int Array of student total Mark.

#### printFirstRow (3 marks)

public static void printFirstRow(int numberOfQuestion)

Print the first row in ID\t\t\tq1\tq2\t.....  
1. Print "ID\t\t\t" (Don't use println).  
2. Loop based on numberOfQuestion, and print "qx\t", where x is 1,2,..., numberOfQuestion  
3. Print a new line.

**Parameters:**

numberOfQuestion - Number of question

#### printAnswerReport (4 marks)

public static void printAnswerReport(String[] studentIds,

int[][] studentAnswers)

Print the answer report  
1. call printFirstRow.  
2. Use nested for loop to print studentIds and studentAnswers. Each result separates with the tab.

**Parameters:**

studentIds - String Array of student id

studentAnswers - 2D int Array of student answer.

#### printWrongAnswerReport (1 mark)

public static void printWrongAnswerReport(String[] studentIds,

char[][] studentAnswers)

Print the Wrong answer report.  
The code is the same as printAnswerReport method since your coding skill is not able to combine it into one at this moment!

**Parameters:**

studentIds - String Array of student id

studentAnswers - 2D char Array of student answer.

#### populateStudentAnswers (3 marks)

public static void populateStudentAnswers(String[] excelList,

String[] studentIds,

int[][] studentAnswers)

Populate/Set studentIds and studentAnswers. Array valve can be updated in another method!  
1. Loop through excelList  
2. Call ExcelHelper.getExcelTo2DStringArray for each item in excelList, and store it in a 2D string array variable.  
3. Call getStudentId, and save it to studentIds.  
4. Call getAllAnswers, and save it to studentAnswers.

**Parameters:**

excelList - String array with all excel files, not including the Answer.xlsx

studentIds - student id string array reference, and it will be filled with student ID.

studentAnswers - studentAnswers 2D int array, and it will be filled with student answer.

#### getStudentId (8 marks)

public static String getStudentId(String[][] answerExcel)

Input 2D Answer Excel and return student ID or "error", if any digit return -1.

**Parameters:**

answerExcel - 2D String Array stores students answer.

**Returns:**

Student ID in String.

#### getDigit (10 marks)

public static int getDigit(String[] row)

Input string array row, and startX, then get the filled digit.  
It is very similar to getAnswerForQuestion!

**Parameters:**

row - String array represents the row of excel.

**Returns:**

0 - 9 (the filled digit) or -1 for missing or filled more than one cell.

#### getAllAnswers (3 marks)

public static int[] getAllAnswers(String[][] answerExcel)

Read all answers from 2D string excel array.  
1. Create int Array "answers" of size 40.  
2. Use for loop to call getAnswerForQuestion for each question, and save it to "answers'  
3. return answers

**Parameters:**

answerExcel - 75 x 26 2D String array

**Returns:**

integer arrays with all answers and each value maps to A to E, and -1 for all other cases.

#### getAnswerForQuestion (15 marks)

public static int getAnswerForQuestion(int q,

String[][] answerExcel)

Read answer to a question from 2D string excel array.  
1. create 2 int variables startX and startY. Check the excel, and set startY to the proper value.   
2. Use if/else to set proper startX according to q, since there are 2 columns!   
3. create char array "allBox" of size 5, and it will store all filled cells.  
4. Loop through all answer related to question q.   
5. Save the first character or ' ', if there is nothing in the cell! (Hints: check length!)   
6. create 2 int variables filledCount and filledPosition.   
7. Loop through "allBox" to count number of answers and save it to filledCount, and store last answer position to filledPosition.   
8. if filledCount and filledPosition are both valid, then return filledPosition, else return -1 to represent error!

**Parameters:**

q - start from 0, and it means question 1 is 0.

answerExcel - 75 x 26 2D String array represent sheet data.

**Returns:**

0 to 4, and it maps to A to E, and -1 for all other cases.

**Assessment scheme**

All Methods passed test cases\* 80 marks

|  |  |
| --- | --- |
| Method Name | Marks |
| readFolderPath | 5 |
| printRow | 4 |
| print2DStringArray | 3 |
| getWrongAnswers | 13 |
| getTotalMarks | 7 |
| printTotalMarkReport | 3 |
| printFirstRow | 3 |
| printAnswerReport | 4 |
| printWrongAnswerReport | 1 |
| populateStudentAnswers | 3 |
| getStudentId | 8 |
| getDigit | 10 |
| getAllAnswers | 3 |
| getAnswerForQuestion | 13 |

Problem-solving technique, e.g. algorithms 5 marks

Java programming technique, e.g. statements, control structures, etc. 5 marks

Programming style, e.g. naming, comments, etc. 5 marks

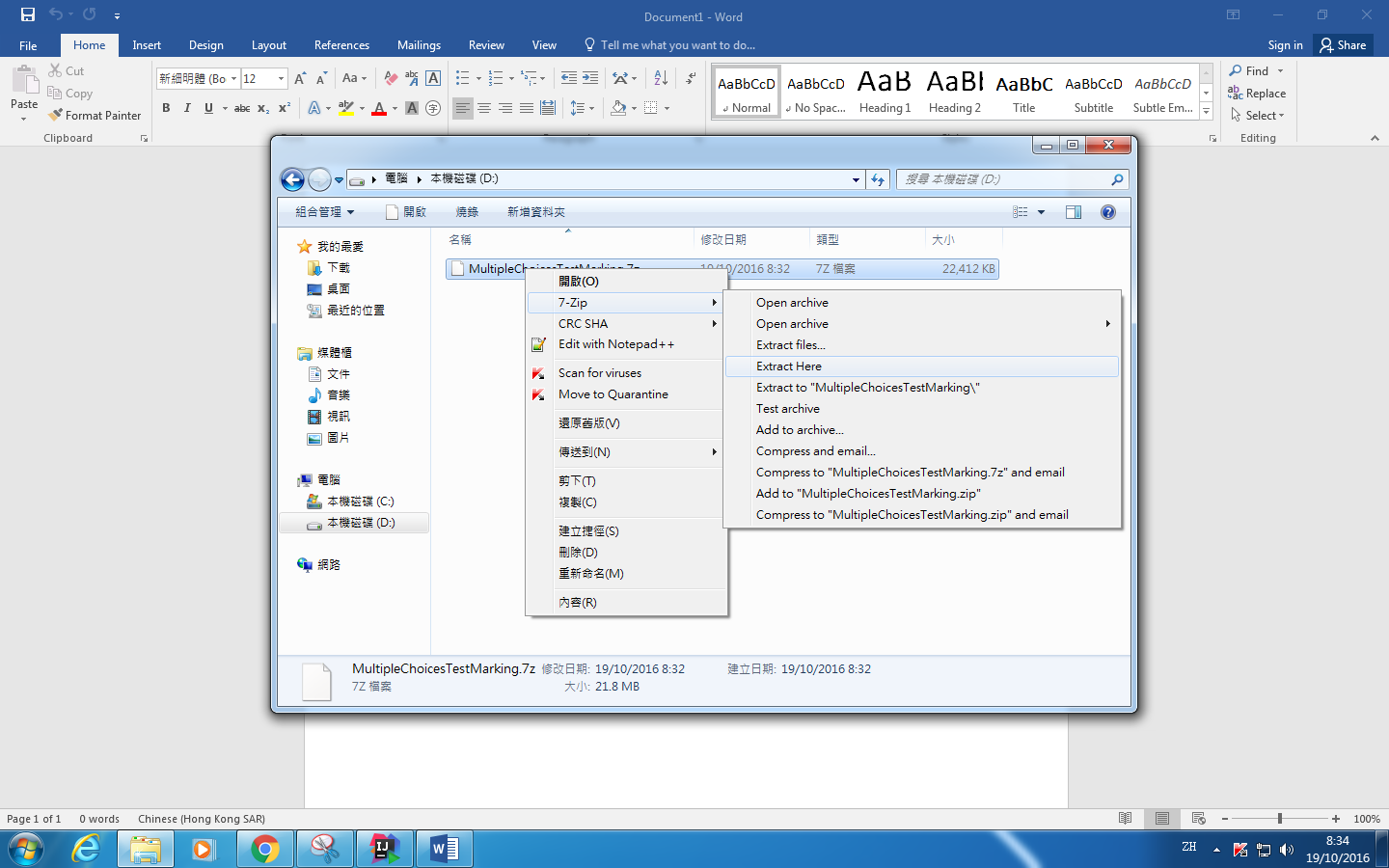
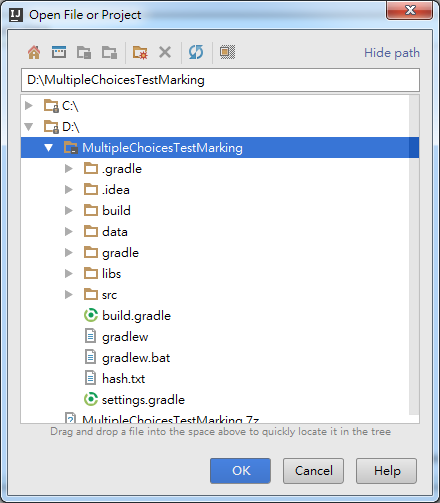
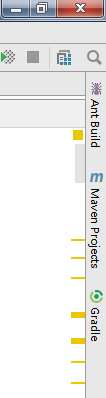
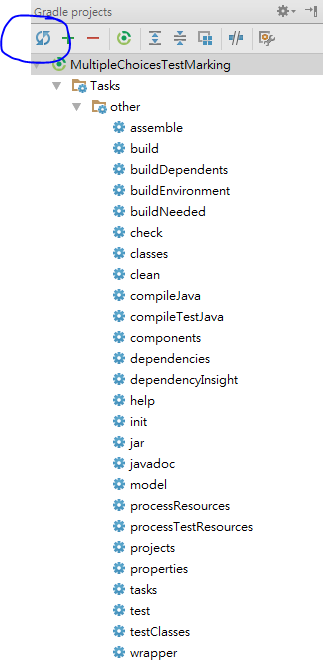
Testing, e.g. Test cases, test results etc. 5 marks

Remarks:

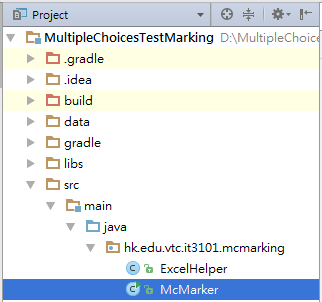
\* There will be additional test cases which will not share to students for marking.

\* You must ensure your code can build most of the time, else you will lose 80 marks!

# Build the project

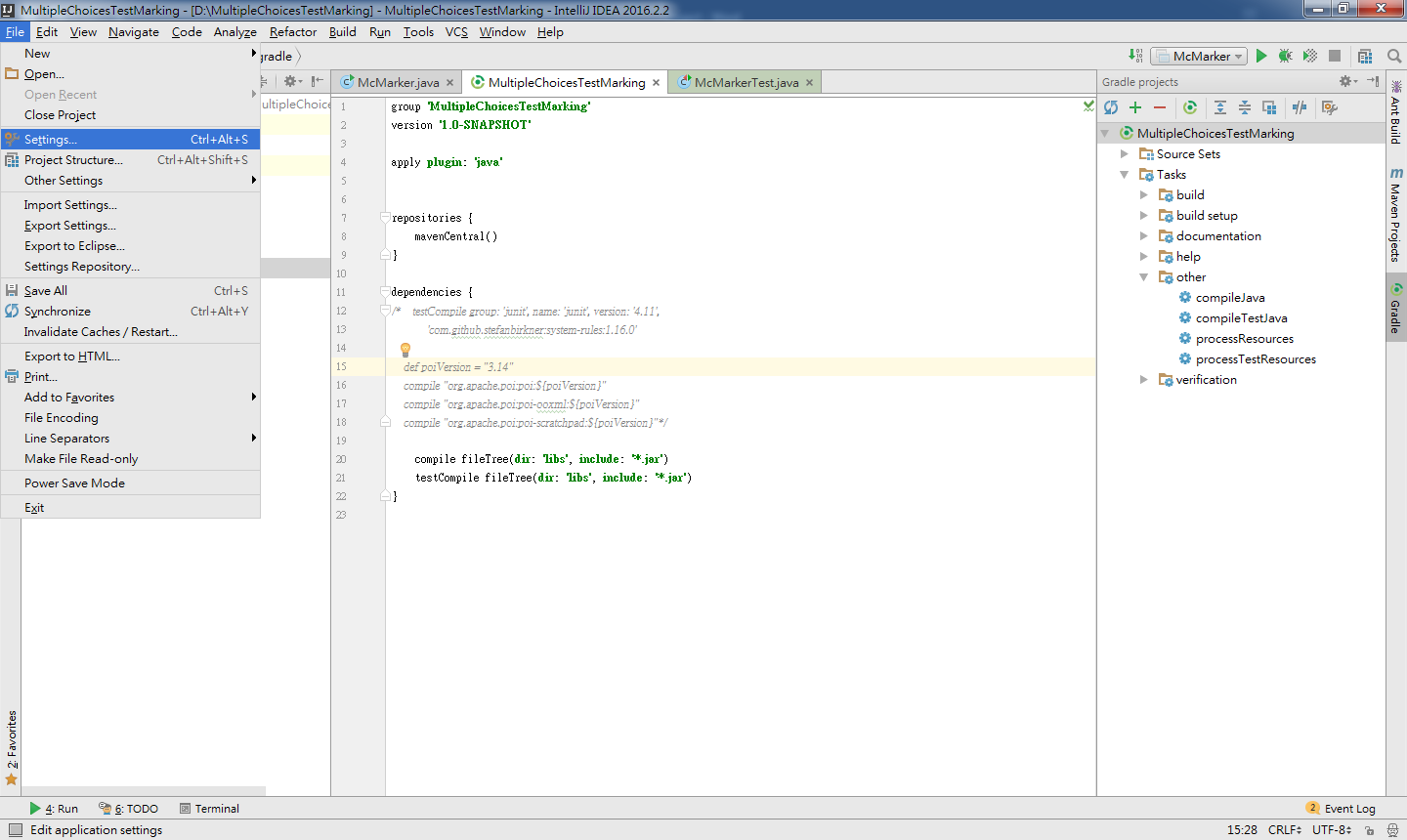
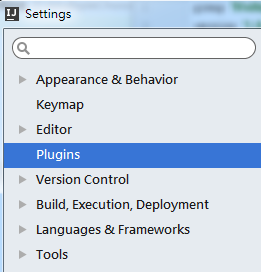
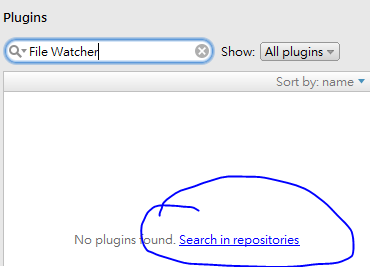
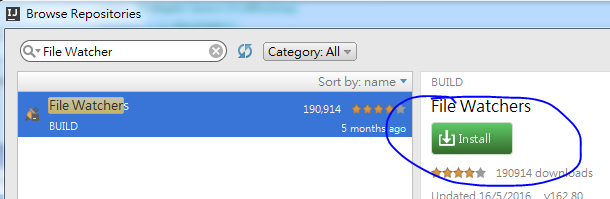
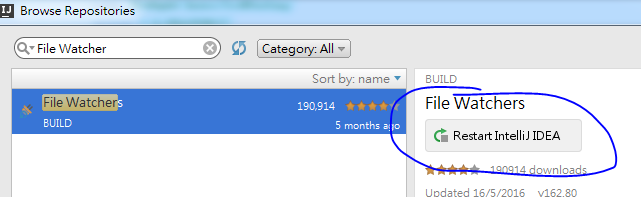
1. Download MultipleChoicesTestMarking.7z and extract here.
2. Follow the same step as **ITE3101 Lab Setup Instruction File.docx** from Moodle.
3. Open the extracted folder.  
   
4. (Optional) Set up JAVA JDK by following the step as ITE3101 Lab Setup Instruction File.docx from Moodle.
5. Click left bottom corner icon, and it will show the hidden panel bar.  
   
6. Click the Gradle tab  
   
7. Click the refresh button, and wait for it complete.  
   

# Replace your assignment key

1. Open McMarker.java   
   
2. Replace REPLACE\_THIS\_WITH\_YOUR\_KEY with your unique assignment key from Email.

|  |
| --- |
| */\* <key>  REPLACE\_THIS\_WITH\_YOUR\_KEY </key> \*/* |

# Install File Watcher Plugin

1. Open Settings (Ctrl + Alt + S)  
   
2. Click on Plugins  
   
3. Type in **File Watcher**, then click **Search in repositories**  
   
4. Click **Install**   
   
5. Click **Restart IntelliJ IDEA**  
   

**--- END ---**