# Technical Training Exercise: Roman Numerals Marking Guide

By Anthony McKale (Wizard without Portfolio)

Code review guidelines <https://github.com/thoughtbot/guides/tree/master/code-review>

|  |  |
| --- | --- |
| MARKS | GENERAL CONDITION RULES |
| AUTO FAIL | If task takes over 4 days |
| AUTO FAIL | If plagiarism |

Each exercise needs a JIRA ticket and a space in SVN created

At the end of each day group all the SVN commits against the JIRA ticket into a Crucible review and review, they will give feedback via Crucible code review tool

Marks are not to be passed on but only for use for comparison purposes, determining how well someone is getting along with the exercise

This exercise is aimed at a mid-level developer level, should be trivial for any senior developer, and any junior developer will struggle.

Purpose of Exercise is to test and improve following skills in a mid-level developer:

-         SVN  
-         Code Review / Jira / Crucible  
-         ES6  
-         (Bonus) ES6 Linting  
-         SCSS  
-         (Bonus) SCSS Linting  
-         Unit Testing  
-         (Bonus) E2E Testing  
-         Aurelia

## Part 1: Download and create Helloworld "Roman Numerals Converter" Application Marking

|  |  |
| --- | --- |
| MARKS | CONDITION |
| AUTO FAIL | If our web-platform is copy and pasted |
| AUTO FAIL | If *node\_modules* is committed to SVN  Or  If Git tags/trunks structure from Aurelia Skeleton Project committed to SVN |
| 4 | For SVN cloning and setting up project |
| 2 | For setup styling, updating package.json title etc |
| 1 | Extracting esnext from Git not using the premade zip  Testable by inspecting package.json  repository/url will equal this value if from premade zip  xxx  bugs/url will equal this value if from premade zip  xxx |

## Part 2: Roman Numerals ES6 class (with Unit Tests) Marking

|  |  |
| --- | --- |
| MARKS | CONDITION |
| AUTO FAIL | no marks if hacked to work  e.g. toArabicNumber(a) {  if(a === 'IV'){  return 4;  }  } |
| AUTO FAIL | If off the shelf Roman Numeral library used |
| 1 | ES6 class RomanNumeralsConverter + ES6 spec |
| 5 | Implement toRomanNumeral(arabicNumber) mark for unit test passed |
| 5 | Implement toArabicNumber(romanNumeral) mark for unit test passed |
| 1 | For using js linting such as "eslint" & creating .eslintrc.json |
| 1 | Implementing Roman Numeral Zero |
| 5 | Implementing Roman Numeral Fractions |
| AUTO EXCERCISE PASS  (Hire immediately!) | Implementing large Roman Numeral Apostrophus or Vinculum systems |

## Part 3: Create Aurelia "roman-numerals-converter" Web Component Marking

|  |  |
| --- | --- |
| MARKS | CONDITION |
| 3 | Creating webcomponent |
| 2 | displays the roman numeral for any number |
| 2 | displays the number for any roman numeral |
| 2 | creating / using view model unit tests |
| 2 | creating / using view model e2e tests |
| 1 | combine both text input fields |

## Part 4: Style Website using SCSS Marking

|  |  |
| --- | --- |
| MARKS | CONDITION |
| 1 | Node Installs of Sass etc |
| 1 | Hooking up Sass to Gulp |
| 3 | Implementing an reasonable design (Sass or CSS) |
| 3 | Using using Sass features sensibly (e.g. nested rules / variables etc) |
| 1 | Using Linting |
| 1 | Linting Passing |