

Undergraduate Programmes - Individual Post-Module Assessment Report

Module code, initials and date: WM164, Jianhua Yang, 17 Feb. 20

Name of marker: Jianhua Yang

Awarded Mark (please note that this does not include late penalties, which will be applied to the mark in Tabula)	
	80

NOTE: The mark recorded is subject to review by the Board of Examiners.

Please note: while each PMA is marked against assessment criteria that are specifically designed to evaluate that assignment, the descriptors listed on the following webpage will help you to interpret the mark awarded to your work:

<https://warwick.ac.uk/services/aro/dar/quality/categories/examinations/markings/ug2017/>

Module Learning Outcomes

The tutor has indicated below if you have achieved the module learning outcomes. They can be "Met", "Partially Met" and "Not Met".

Learning Outcome	Status
Know the concept of variables, numerical operators and the basic methods of formatting and displaying data using a programming language. Know the differences between logical, bitwise and relational operators.	Met
Use loops for coding the repetitive tasks used in software development of automobiles.	Met
Use decision structures to select the right function to be enabled for a specific set of input given by customer.	Met
Use in-built and user-defined functions for coding programs according to customer requirements.	Met
Implement the concept of exception handling in programs. Comprehend the fundamental data structure types.	Met

Please note: some module learning outcomes may be addressed by other assessments or in-module work.

Individual Comments

The tutor has provided specific comments on your work below and they may have provided an annotated script.

Strengths & Areas for Improvement
Good use of branching for source control. A number of functionalities were provided by the programme. Source code follows the PEP8 style in general. Good use of function to group programme logics together. A number of 3rd party libraries were used.
How to improve
You could try to implement OOP. Also, try to arrange code into separate modules to show the structure and interaction of different components i.e. the architecture. Docstrings were only partially implemented. No formal testing - a number of user testings were reported, but not in the sense of test-driven development.
Document structure, grammar, referencing
Good use of formatting and styles. Very good to include flow charts and pseudocode. However, figures should have captions and be referred to in the main body of the text.
Anything else
Academic style of writing recommends a minimum use of 1st person, even though not absolutely forbidden.

