

# WARWICK IFP – COMPUTER SCIENCE

## PYTHON3 PROGRAMMING PROJECT (30%)

### SUMMATIVE 02

#### Dates:

**Release:** 2 March 2020 MONDAY 12pm – Information provision

**Closed Assessment:** w/9 March 2020 in double seminar sessions

**Final Submission:** 27 April 2020 MONDAY 12pm

Total time (including Easter): 8 weeks

#### TASK

You need to demonstrate skill and application of the Python programming language. Summative Assessment 2 is in two parts.

1. Closed Assessment - A program written in Python
2. Project write up

#### Closed Assessment

**Programming Task:** w/9 March 2019

Students are given a programming task, for which they will have 2 hours to develop a solution and write a Python program. The completed program will be submitted via Tabula at the end of closed assessment. The program will contribute a  $\frac{1}{2}$  (15% of the 30%) of summative assignment 2.

#### Project Analysis

**Final Submission:** 29 April 2019

You will write a report that reflects upon the process you undertook to develop your solution to the programming task.

Submit **exactly** one file (Microsoft Word or PDF format):

1. A 1000-word program evaluation report

#### OBJECTIVES BEING TESTED

**Productive Skill:** You need to demonstrate skill and application of the Python programming language to solve a real computational problem. For this, you must submit a working program that presents a range of programming techniques. The task will only test the skills and knowledge you have learned during the module. You will complete a 2-hour closed programming task, and subsequently you will submit a program that implements your solution.

Critical Thinking Skill: You need to demonstrate your ability to evaluate the effective delivery of Python programs. Your 1000-word report will describe your evaluation of a piece of software that solves a problem and areas for improvement. You will need to describe the program's implementation and indicate the programming techniques and areas of subject knowledge demonstrated.

## DETAILS

### Programming Part:

Implement your solution to your computational problem in **Python 3.6 or later**, using any relevant in-built Python libraries. You must use comments in your program to assist an unfamiliar reader with your code. Test that your program will execute before submission.

You will have access to Tabula and the Python Documentation from <https://docs.python.org/3/>. You can use **PyCharm** to support your development process but it is not a requirement.

### Report Parts:

Write a 1000-word analysis and design report of your Python program. Describe how your solution solves the problem and identify the evidence you gained through testing. You are encouraged to include diagrams to illustrate your solution, and small snippets from your code. You can use excerpts from your program and they will **not** be included in your word count. Your report should identify the quality of the solution with respect to the information given, and the programming techniques that have been applied and tested in Python 3.

## INDIVIDUAL SUBMISSION

The problem selected and the computational solution given must be your own work and it must not be influenced by projects being developed by other students. Students are referred to the university rules regarding plagiarism and cheating.

[https://warwick.ac.uk/services/aro/dar/quality/categories/examinations/policies/i\\_suspectedcheating](https://warwick.ac.uk/services/aro/dar/quality/categories/examinations/policies/i_suspectedcheating)