Nottingham Trent University

School of Science and Technology

COURSEWORK ASSESSMENT SPECIFICATION

Nottingham Trent University

Module Code	COMP10082
Module Title	Foundations of Computing and Technology, programming strand
Module Leader	Nuno Amálio
Module Team	Nuno Amálio Ismahane Cheeheb Azi Meskaran Matthew Harris Abdallah Naser Thomas Johnson Salim Maaji Amna Anwar
Coursework Title	Sueca card game scorer
Module Learning outcomes assessed	Write python Programs providing computing solutions to real-world problems
Contribution to element	90%
Date set	05/12/2022 at 14:30
Method of Submission	Via the NOW system, according to the instructions provided therein
Feedback Provision	Week commencing 16th of January
Method of Feedback	Feedback to be delivered on NOW

Work handed in up to five working days late will be given a maximum Grade of Low Third whilst work that arrives more than five working days will be given a mark of zero.
Work will only be accepted beyond the five working day deadline if satisfactory evidence, for example, an NEC is provided.
https://www.ntu.ac.uk/studenthub/my-course/student-handbook/submit-a-notification-of-extenuating-circumstances

The University views **plagiarism and collusion** as serious academic irregularities and there are a number of different penalties which may be applied to such offences. The **Student Handbook** has a section on Academic Irregularities, which outlines the penalties and states that **plagiarism** includes:

'The incorporation of material (**including text, graph, diagrams, videos etc.**) derived from the work (published or unpublished) of another, by unacknowledged quotation, paraphrased imitation or other device in any work submitted for progression towards or for the completion of an award, which in any way suggests that it is the student's own original work. Such work may include printed material in textbooks, journals and material accessible electronically for example from web pages.'

Whereas **collusion** includes:

"Unauthorised and unacknowledged copying or use of material prepared by another person for use in submitted work. This may be with or without their consent or agreement to the copying or use of their work."

If copied with the agreement of the other candidate both parties are considered guilty of Academic Irregularity.

Please remember submitting portions of work already assessed is **Self-Plagiarism** and is also a serious academic irregularity.

Penalties for Academic irregularities range from capped or zero grades for elements of modules, to dismissal from the course and termination of studies.

To ensure that you are not accused of plagiarism, look at the sections on **Plagiarism Support and Turnitin support.**

I. Assessment Requirements

The coursework of the programming strand of Foundations of Computing Technology will be about developing a python program, called *sueca_scorer*, which: (i) calculates scores of given sueca card games encoded in text files, and (ii) detects situations of invalidity and non-compliance to the rules of the Sueca game.

Submissions may be undertaken individuals, or by students working in pairs. By default, submissions are individual. Coursework pairs must be registered and are subject to vetting to ensure that pairs are even with respect to programming ability; students of a pair must be in the same class.

II. Assessment Criteria

The next table presents the distribution of marks across the different parts of the python implementation of *sueca_scorer*, highlighting the focus of the marking with respect to what is described in the coursework description. The mentioned sections of the coursework description prescribe either specific implementation definitions, namely, functions, classes and exceptions, or describe desired features of the implementation.

Implementation Part	Marks	Focus
Ranks and suits	7	Definitions of section 3.1 of the coursework description are worth the following marks: • Functions ‘valid_suit’, ‘valid_rank’, and ‘suit_full_name’ (3 marks) • Function ‘rank_points’ (1.5 marks). • Function ‘rank_higher_than’ (2.5 marks).
Cards	8	Definitions of section 3.2 of the coursework description are worth the following marks: • Exception ‘CardInvalid’ (1 mark) • Function ‘parseCard’ (1 mark) • Class constructor (1.5 marks) • Class function ‘points’ (1 mark) • Class function ‘higher_than’ (2.5 marks) • Class function ‘show’ (1 mark)
Tricks	10	Definitions of section 3.3 of the coursework description are worth the following marks: • Function ‘parseTrick’ (1.5 mark). • Function ‘parseGameFile’ (3 marks). • Class constructor (1.5 mark). • Class function ‘points’ (1.5 marks). • Class function ‘trick_winner’ (1.5 marks) • Class function ‘show’ (1 mark)
Games	14	Definitions of section 3.4 of the coursework description are worth the following marks: • Exceptions ‘CardAlreadyPlayed’, ‘DealerDoesNotHoldTrumpCard’ and ‘IllegarCardPlayed’ (3 marks) • Class constructor (1 mark) • Class functions ‘gameTrump’, ‘score’ and ‘gameTricks’ (3 marks) • Class function ‘playTrick’ (5 marks) • Function ‘cardsOf’ (2 marks)
Main module and	11	Definitions and descriptions of sections 2 and 3.5

command-line Interface		of are worth the following marks: • Exceptions ‘GameFileCouldNotBeFound’ and ‘SuecaGameIncomplete’ (2 marks). • Function ‘runGame’ (4 marks). • Command-line interface and outputs described in section 2 (5 marks).

The next table, which maps coursework marks to NTU grades, is indicative. Because a new system of automated marking is being put in place this academic year, some adjustments to the mapping may take place for pedagogical reasons, to better reflect the merit of submissions, if the module team deems that such an adjustment is necessary for the sake of the fairness and integrity of the grading.

Coursework marks	**Grade**
47–50	Exc 1st
43–46	High 1st
39–42	Mid 1st
35-38	Low 1st
34	High 2:1
32-33	Mid 2:1
30-31	Low 2:1
29	Mid 2:2
27-28	Mid 2:2
25-26	Low 2:2
24	High 3rd
22-23	Mid 3rd
20–21	Low 3rd
18–19	Marginal fail
15–17	Mid Fail
1–14	Low Fail
0	Zero

III. Feedback Opportunities

Coursework grades and feedback will be released three working weeks after the submission, which is expected to take place in early January.

Students are advised to to take advantage of the support being provided in the module's labs and surgeries to get both feedback and support.

IV. Resources that may be useful

Referencing styles please use Harvard as detailed here
Guide to planning your time here and an automated planner here
Guidance on avoiding cheating is here
Remember to use Outlook or physical calendars to block out time between lectures and labs to work on this coursework.

V. Moderation

The Moderation Process

All assessments are subject to a two-stage moderation process. Firstly, any details related to the assessment (e.g., clarity of information and the assessment criteria) are considered by an independent person (usually a member of the module team). Secondly, the grades awarded are considered by the module team to check for consistency and fairness across the cohort for the piece of work submitted.

VI. Aspects for Professional Development

The assessment helps students to enhance and evaluate their python and general programming skills, which are valued by employers and at the core of any computing degree.