

NOTTINGHAM TRENT UNIVERSITY



School of Science and Technology

COURSEWORK ASSESSMENT SPECIFICATION (UG)

Details of Module and Team

What Learning Outcomes are assessed?

What are my Deadlines and how much does this assessment contribute to my Module Grade?

What am I required to do in the assessment?

What are my assessment criteria? (What do I have to achieve for each grade?)

Can I get formative feedback before submitting ?
If so, how?

What extra support could I look for myself?

How and when do I submit this assessment?

How and when will I get summative feedback?

What skills might this work evidence to employers?



2

MODULE CODE	COMP10082
--------------------	-----------

MODULE TITLE	Foundations of Computing & Technology - Programming
MODULE LEADER	Dr Amin Safaei
TUTOR(S)	Amin Safaei Jake Street Salim Maaji Jake Street Thomas Johnson Nida Zeeshan Gulrukh Turabee Matthew Harris Nida Zeeshan Matthew Harris Sean Haddick
COURSEWORK TITLE	Programming Class Test & Project
LEARNING OUTCOMES ASSESSED	
CONTRIBUTION TO ELEMENT	(1 of 2) 50% of module coursework
DATE SET	Teaching week 4
DATE OF SUBMISSION	<ul style="list-style-type: none"> ✓ Class Test – Online during Week 8 (Monday 15th Nov 2021) ✓ Project – 9am on Monday 6th December 2021 ✓ Demonstration of working code - week beginning 6/12/2021
METHOD OF SUBMISSION	Class Test - Online Project -via dropbox
DATE OF FEEDBACK	Class Test : 7 days after submission Project :21 days after submission
METHOD OF FEEDBACK	Mark and feedback provided electronically
<p>Work handed in up to five working days late will be given a maximum Grade of a Low Third, whilst work that arrives more than five working days late will be given a mark of zero.</p>	

Work will only be accepted beyond the five working day deadline if satisfactory evidence, for example, an NEC is provided. Any issues requiring a NEC can be found at:

https://ntu.ac.uk/current_students/resources/student_handbook/appeals/index.html

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*"The incorporation of material (**including text, graph, diagrams, videos etc.**) derived from the work (published or unpublished) of another, by unacknowledged quotation, paraphrased imitation or other device in any work submitted for progression towards or for the completion of an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.

Penalties for Academic irregularities range from capped marks and zero marks 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 assessment process for this strand is based on 2 parts:

5

1. a **class test** and,
2. a **program of your own choice** (this has to be agreed upon with your tutor).

II. Class Test

The **class test** will comprise of **an online programming test** where you will be given a problem and expected to code a solution to it within 90 minutes (with extra time allowed if you have been authorised).

The test will be broken into 3 sections. The test will require using sequences, selections, iteration, functions as well as anything that has been taught.

The test will be worth 30% of the mark for the programming strand. The mark will be awarded at 10% for each completed part.

The test is open book – access to notes, textbooks and internet will be allowed.

If you pass the class test, you will be allowed to proceed to your programming project based on the idea that you will have discussed with your tutor before the test takes place.

If you do not pass the class test, you will not be allowed to proceed to the project. Instead, you will be required to sit a referral class test in week 10. This class test will allow you to score a maximum of a 3LOW and pass the module. You will be given details closer to the time on the arrangements for this by the module leader.

III. Programming Project

If you have passed the class test, you will be allowed to proceed to the programming project. You will design and implement a piece of software using Python. The required functionality of this piece of work **must** be agreed and signed off with the teaching lab tutor during week 6 (1st – 5th November 2021), otherwise a fixed task will be assigned to you.

The signoff process is as follows:

- You must discuss the proposed project with your PTL tutor.
- If your tutor is satisfied that the project is appropriate, you must write a short specification stating the aims and outcomes (<300 words) of the project.
- This needs to be uploaded to the “**Project Details**” dropbox by the end of week 5 .
- Your tutor will read through the details and reply with a **Yes** or **No** feedback in dropbox during week 6.
- If the answer is **no**, you will need to discuss this with your tutor to find out how to expand or change your idea. Once the changes/new idea has been agreed, re-upload the specification for your tutor to look at again and sign off.

The code must be designed to achieve a goal – it cannot be simply a set of sections of code to demonstrate aspects of Python.

The program should show evidence of at least the basic use of loops, branching, data structures and file handling and any further advanced features that might be needed to achieve the goal of the program.

The mark for this project (**the remaining 70% for the programming strand**) will be mainly derived from a demonstration of the project which will include a question and answer session where your tutor will ask you about the design and structure of your project. Documentation will be used to help in this process. If you do not attend your demonstration, or, the project shows inadequate ownership of the presented code, your tutor may award a mark of zero.

Your implementation may use existing libraries or other code, provided they are fully attributed. A lack of attribution will be considered as plagiarism and the usual action will be taken for these cases. Please ensure that you read the student handbook regarding how to cite and reference, as well as look at the materials available on the library web page. In addition, the work needs to be done by you and you must not work with others on any code. The contribution of the student relative to the amount of code they have sourced from elsewhere will be assessed when awarding the mark.

The documentation you will need to submit in addition to all source code, should be as follows:

- Brief description of the **aims** of program
- **Analysis** of the requirements of the program (1-2 pages)
- **Design** – may be documented by flow charts, class diagrams etc. (2-4 pages)
- **Testing** – table of tests to be undertaken to prove that the program achieves its goals (1-2 pages)
- **Critique** – what worked, what didn't and what could have been improved (up to 1 page)

II. Assessment Criteria

Marking criteria	Exceptional first / 1	2:1	2:2	3	Fail	7ED0
Project Implementation & Report	The program demonstrates an excellent knowledge and understanding of programming. Excellent design and well-thought out implementation, using sophisticated constructs, complexity and structure where appropriate. Shows effort/learning that is well beyond taught material. The program achieves all of the project aims.	The program demonstrates very good knowledge and understanding of writing more complex programs. Good design and implementation. The complexity and structure of the program is of a good standard and the program performs very well and accomplishes most of the aims.	The program demonstrates good knowledge and understanding of programming and achieves the aims expected for a student who has engaged with the module. The program provides some complexity and structure. The design relates to how the program works.	The project demonstrates that knowledge and understanding is sufficient to deal with basic terminology, concepts and programming. Overall structure and complexity is limited and meets the basic aims of the program. The student has demonstrated that they achieved some programming skills and knowledge but are not fully able to apply these to more complex systems. The design is limited and does not go into much detail regarding how the program works.	Insufficient knowledge and understanding of the area. The program does not do much and shows limited understanding of programming. Little design will be visible because of the limited nature of the aim of the project.	W A N T I N G T O B E A S E D O N T P A S S

The Class Test

This will be given by Boolean marks for each section which will indicate either a pass or fail. For a section to be passed, it has to achieve the goal specified in the rubric using standard programming techniques e.g. loops or any other programming constructs and not by simply outputting the answers as determined by calculating by hand.