

Introduction to Programming Assessment (DAP403)

General Assessment Guidance

- The summative assessment for this module is coursework submission which accounts for 100% of the marks.
- The deadline for submission is 14 January 2022 1:00 pm, and the submission word limit is 2000 words.

The Assessment Scenario:

Chi Cleaners is a dry-cleaning service provider. The owner is looking to automate their point of sale operation and has asked you to design and develop a prototype program for a single customer showcasing the point of sale program to be used by staff.

Chi Cleaners offers two types of services (standard, specialist) in two categories, Ladies and Gentlemen, as shown in Table 1. Customers are provided with a receipt that shows the total number of items and the total cost of services with separate VAT calculated at 20%. The customers receive a discount of 10% on orders over £15 and 15% on orders over £30 (see [Charging VAT Guidelines](#)).

They also need to print and attach small tags to every piece of the customer's item to be identified during the dry-cleaning process. They use a 12-digit numbering system, a combination of date in the format of [dd/mm/yyyy], time in the format of [hh:mm], and type of requested service. For example, if the order took place on 11/03/2021 at 14:22, the printed tag should be 110320211422-sp, where [-st] stands for standard and [-sp] stands for specialist services.

Ladies			Gentlemen		
	Standard	Specialist		Standard	Specialist
Dress	£8.00	£11.20	2 PC Suit	£12.00	£16.80
Eve Dress	-	£21.00	2 PC Eve Suit	-	£16.80
2 PC Suit	£12.00	£16.80	Jacket	£7.50	£10.50
Jacket	£7.50	£9.10	Trousers	£6.50	£9.10
Shirt	£6.50	£9.10	Shirt	£6.50	£9.10
Blouse	£6.50	£9.10	Coat	£9.95	£13.93

Table 1: *Chi Cleaners service/price guide*

General Requirements

- You are required to design and develop a procedural Python program (not OOP) using only Python's standard libraries.
- You are required to comment single source code file adequately.
- The Python console should act as an interface to interact with the point of sale program.
- If the operator enters invalid inputs, the program should not terminate, and a relevant message should help the operator proceed with appropriate inputs.
- If you think the provided information in the scenario is insufficient, you need to make justified assumptions for the design and development of the point of sale program.

Tasks

Task 1: Customer Receipt

You are required to design and develop a program as a point of sale operation for a single customer, given the information provided in the scenario. The program should prompt the operator with the necessary options to guide them through selecting items/services and processing a receipt. Finally, the program should create an external file called [receipt.txt] with the details of the customer's items, services and costs.

Task 2: Tagging

You are required to design and develop a tagging program that prints tags for every piece of a customer's item, reflecting the item's name, type of service and tagging system. The tags should have enough margins and guides for cutting and should appear in an external file called [tag.txt].

Task 3: Testing

Create, document and conduct suitable testing, ensuring the outputs are correct and invalid input values are handled correctly.

Task 4: Documentation (report)

You are required to produce program documentation of no more than 2000 words (excluding references and appendices) for your program and should include the following sections:

1. Introduction

Explain the problem and an overview of the solution. Also, provide the details and justifications of assumptions due to insufficient information in the scenario (requirements).

2. Program Structure, Input and Output

Explain in detail the algorithm of the program using a flowchart for the requirements of **Task 1** and **Task 2**, covering the following:

- Inputs and outputs of the program and their types
- Functions, modules, calculations and variables

3. Testing

Create, document and conduct suitable testing, ensuring the outputs are correct and invalid input values are handled correctly.

4. Appendices

Copy of source code as text (not screenshots)

Deliverables

Part 1. Documentation named with your student number (1234567.pdf)

Part 2. Single Python source code file named with your student number (1234567.py)

Notes:

1. Python source code should run without errors on the university machines and produce the reported output. Otherwise, you will fail the assessment (you will not be assessed on the quality/performance of source code).
2. You are required to submit both parts for a valid submission. Otherwise, you will fail the assessment

Assessment Criteria

Assignment Parts	Approach	Mark
Part 1: Python Program & Documentation	Python Program	60%
	70 and above Fully working program, free from syntax and logical errors, covering all aspects of the requirements. Making use of excellent programming practices, structures and well-commented code.	
	60-69 Fully working program, free from syntax and logical errors, covering most aspects of the requirements. Making use of good programming practices and structures with commented code.	
	50-59 Fully working program, free from syntax errors, some logical errors, covering some aspects of the requirements. Making use of fair programming practices and structures with fairly commented code.	
	40-49 Fully working program, free from syntax errors, many logical errors, covering a few aspects of the requirements. Basic use of programming practices and structures with minimally commented code.	
	<40 (fail) A non-functional program that produces syntax errors, minimal or no attempt to address the requirements. Poor programming practices and structures with no or little commented code.	
	Documentation	
	70 and above An excellent, analytical, detailed and well-written report covering all the required sections.	
	60-69 A good, detailed and well-written report covering most of the required sections.	
	50-59 A descriptive with some details report, covering some of the required sections.	
	40-49 A descriptive report covering a few required sections.	
	<40 (fail) A poorly written report and required sections are not covered.	

Part 2: Testing	<p>70 and above</p> <p>Evidence of relevant and comprehensive testing, demonstrating an excellent understanding of testing procedures.</p> <p>60-69</p> <p>Evidence of relevant and comprehensive testing, demonstrating a good understanding of testing procedures.</p> <p>50-59</p> <p>Evidence of some testing, demonstrating some understanding of testing procedures.</p> <p>40-49</p> <p>Minimal evidence of testing, demonstrating only a basic understanding of testing procedures</p> <p><40 (fail)</p> <p>Poor or no testing.</p>	<p>35 %</p>
Structure and Referencing	<p>All the sections/subsections must be numbered except References and Appendices. All the figures/tables in the report must have captions (e.g. Figure 1, Table 1), adequately explained and referenced.</p> <p>Appropriate use of referencing and structuring report with numbered sections, titles, subtitles.</p> <p>The language should be appropriate to the academic context and business communication.</p> <p>Avoid personal language:</p> <p>Academic writing is impersonal. You should not use words like I, my, me, myself etc.</p> <ul style="list-style-type: none"> • use a passive rather than an active verb. • make words such as 'the report', 'this section' the subject of the sentence. 	<p>5%</p>
	<p>Total</p>	<p>100</p>