



**FOUNDATION ASSESSMENT II MATERIAL RELEASE**

THEORY QUESTIONS

| **SECTION** | **MARK** |
| --- | --- |
| **Theory Questions** | 31 |
| **Concept Questions** | 19 |
| **Python Challenge** | 25 |
| **SQL Challenge** | 25 |
| **TOTAL** | **100** |

**Important notes:**

* This document shares the first section of the Foundation Assessment II which is composed of 9 Theory Questions
* It is worth just under a third of your assessment mark
* You have 24 hours before the assessment to prepare.
* If any plagiarism is found in how you choose to answer a question you will receive a 0 and the instance will be recorded. Consequences will occur if this is a repeated offence. You can remind yourself of the plagiarism policy [here](https://drive.google.com/file/d/1k9UaGOR7hx54QRZ8jvp2jtC4P-8_Rs4F/view?usp=sharing).

**Section 1: Theory Questions [31 marks]**

| **1.1 What does SDLC stand for?**  **ANSWER: SDLC stands for Software Development Life Cycle** | **1 mark** |
| --- | --- |

| **1.2 What exception is thrown when you divide a number by 0?**  **ANSWER: In Python dividing by zero raises a ZeroDivisionError** | **1 mark** |
| --- | --- |

| **1.3 What is the git command that moves code from the local repository**  **to the remote repository?**  **ANSWER: git push** | **1 mark** |
| --- | --- |

| **1.4 What does NULL represent in a database?**  **ANSWER: This indicates that there is no value assigned to that field, data does not exist or has not been provided.** | **1 mark** |
| --- | --- |

| **1.5 Name 2 responsibilities of the Scrum Master**  **ANSWER: Monitoring and Reporting of the team**  **Enabling Scrum Events e.g. daily stand-ups and sprint reviews** | **2 marks** |
| --- | --- |

| **1.6 Name 2 debugging methods, and when you would use them.**  **ANSWER: Step-through executes code line-by-line enables changes in variables to be noticed.**  **Exception Handling handles and catches errors that occur during the execution of a program** | **4 marks** |
| --- | --- |

| **1.7 Looking at the following code, describe a case where this function**  **would throw an error when called.** Describe this case and talk about  what exception handling you’ll need.   | **def can\_pay(price, cash\_given):**  **if cash\_given >= price:**  **return True**  **else:**  **return False** | | --- |   **ANSWER: There is a possibility of experiencing a runtime error if ‘price’ variable and ‘cash\_given’ variable are not well-suited for each other e.g. one is given as a string and the other is given as a numerical value. Running the program would give a ‘TypeError’.**  **I would use a try and except block in my code to catch the exception e.g. except TypeError then return False** | **5 marks** |
| --- | --- | --- |

| **1.8 What is git branching?** Explain how it is used in Git.  **ANSWER: Git branching is a feature of Git, it is a distributed version control system. Git branching enables collaboration and parallel development. Branching enables programmers to fork from the main development and create separate places to work. This is commonly used to work on multiple different features as well as fix bugs concurrently without disturbing other team members' code. Using git command, allow for completed branches to be merged back into the main line of workflow ‘git merge’ , and for programmers to switch between different branches ‘git checkout’.** | **6 marks** |
| --- | --- |

| **1.9 Design a restaurant ordering system.**  You do not need to write code, but describe a high-level approach:   * 1. Draw a list of key requirements   2. What are your main considerations and problems?   3. What components or tools would you potentially use?   **ANSWER:**   1. Draw a list of key requirements   **Menu Management: The system should permit the restaurant to manage their menu, by allowing adding/updating and removing menu items.**  **Customer order placement and processing: Customers should be able to look through the menu, select their item/items of choice, have the possibility to customise their order and finally place the order.**  **Payment method and transaction storage: Ensure payments are secure and transactions are kept safely.**  **Levels of Access/ Permissions: The system should issue different administrative access for different members of staff e.g. a manager would have different access rights to that of a kitchen staff member.**   1. What are your main considerations and problems?   **Scalability of the system: As always restaurants are dealing with high demand of order intakes dependent on the restaurant and how many branches of the restaurant the system should be able to deal with high volumes of orders especially during busy times.**  **User Experience (UX): Ensure a users-friendly interface for the customers as well as staff to ensure that the customers have an enjoyable experience.**  **Backups: The system to be reliable and regularly have a set automated backup in place, to ensure that in the case of data breach data can be recovered and minimise the downtime of the restaurant.**   1. What components or tools would you potentially use?   **Database: A (RDMS) relational database management system e.g. MySQL in order to store the menu items, customer information as well as keep track of orders.**  **API Development: Using RESTful APIs to help support Menu Management like adding, removing and updating menu items.Furthermore, providing additional information about what’s offered e.g. vegetarian, vegan or allergens**  **Security Measures: Applying digital signatures and certificates as well as using newer encryption protocols to ensure that customers and employees information are transmitted and kept securely.** | **10 marks** |
| --- | --- |