



**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. **What does SDLC stand for?** | **1 mark** |

SDLC stands for Software Development Life Cycle.

|  |  |
| --- | --- |
| **1.2 What exception is thrown when you divide a number by 0?** | **1 mark** |

When you divide a number by zero in most programming languages, including Python,

it raises a "ZeroDivisionError" exception.

This exception indicates that the division operation attempted to divide a number by zero, which is mathematically undefined.

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

git push

|  |  |
| --- | --- |
| **1.4 What does NULL represent in a database?** | **1 mark** |

NULL represents the absence of a value or the lack of a value in a particular field or column.

It is a special marker used to indicate that a data entry is missing, unknown, or not applicable.

|  |  |
| --- | --- |
| **1.5 Name 2 responsibilities of the Scrum Master** | **2 marks** |

As the Scrum Master, there are several responsibilities to fulfill in order to support and facilitate the successful implementation of the Scrum framework. Here are two key responsibilities of the Scrum Master:

Facilitating Scrum Events: The Scrum Master is responsible for facilitating various Scrum events or ceremonies. These events include the Sprint Planning meeting, Daily Scrum (Daily Standup), Sprint Review, and Sprint Retrospective. The Scrum Master ensures that these events are effectively organized, timeboxed, and focused on achieving their intended goals. They facilitate the collaboration and communication among the Scrum Team members during these events, helping to ensure that everyone is aligned and working towards the Sprint's objectives.

Removing Impediments: Another important responsibility of the Scrum Master is to identify and remove any impediments or obstacles that may hinder the progress of the Scrum Team. The Scrum Master acts as a servant-leader, working closely with the team to understand any challenges they face and helping to address them. This involves proactively identifying potential issues, facilitating discussions to find solutions, and working with stakeholders to remove any roadblocks that could prevent the team from delivering the Sprint Goal.

|  |  |
| --- | --- |
| **1.6 Name 2 debugging methods, and when you would use them.** | **4 marks** |

Logging:

Logging is a widely used debugging method that involves adding code to write informative messages to a log file or console during the execution of a program. These log messages can help track the flow of execution, identify errors, and understand the state of variables or data at different points in the code. Logging is useful when you want to investigate an issue that occurs during runtime or when you want to gather information for later analysis. By reviewing the log messages, you can trace the program's execution path and identify the cause of errors or unexpected behavior.

Debugging Tools and Breakpoints:

Debugging tools and breakpoints are essential for interactive debugging. Integrated Development Environments (IDEs) often provide debugging features that allow you to set breakpoints at specific lines of code. When the program execution reaches a breakpoint, it pauses, allowing you to inspect the program's state, variable values, and step through the code line by line. This method is particularly useful when you need to identify the specific line of code that is causing an error or unexpected behavior. It allows you to observe the program's execution in real-time and pinpoint the source of the issue.

These methods can be used in combination or individually, depending on the nature of the problem and the tools available. Logging is helpful for capturing information and analyzing issues that occur during runtime, while debugging tools and breakpoints provide a more interactive and detailed approach to examine the program's execution and identify the root cause of a problem.

|  |  |  |
| --- | --- | --- |
| **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** | | **5 marks** |

def can\_pay(price, cash\_given):

try:

if float(cash\_given) >= float(price):

return True

else:

return False

except ValueError:

# Exception handling for non-numeric values

return False

|  |  |
| --- | --- |
| **1.8 What is git branching?** Explain how it is used in Git. | **6 marks** |

Create a New Branch: A new branch can be created from an existing branch (often from the main branch) using the command git branch <branch\_name>

Switch to a Branch: Developers can switch between branches using git checkout <branch\_name>.

Commit Changes

Merge Branches: Once the work on a branch is complete and tested, the changes can be merged into another branch (e.g., the main branch) using git merge <branch\_name>.

Resolve Conflicts: If there are conflicting changes between branches being merged,

Delete Branches: Branches that are no longer needed can be deleted using git branch -d <branch\_name>. This removes the branch and its commits from the repository.

|  |  |
| --- | --- |
| **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? | **10 marks** |

Menu Management: The system should allow restaurant staff to manage the menu, including adding, updating, and removing menu items.

Table Management: The system should support table management, allowing staff to assign tables to customers and track their orders accordingly.

Customization and Special Requests: Customers should be able to customize their orders and make special requests, such as specifying dietary preferences or allergies.

Payment Integration: Integration with various payment gateways or payment methods to facilitate secure and convenient payment processing.

Reporting and Analytics: The system should provide reporting and analytics features to help track sales, inventory, popular items, and other relevant data for business analysis.

b. Main considerations and problems:

User Experience: Ensuring a seamless and user-friendly ordering experience for both customers and restaurant staff is essential.

Scalability: The system should be designed to handle a large volume of orders, especially during peak hours, without performance issues.

Integration: Integration with existing systems such as POS (Point of Sale), inventory management, or kitchen display systems may be necessary.

Security: Protecting customer data, payment information, and ensuring secure transactions is a critical concern.

Communication: Efficient communication between customers, kitchen staff, and serving staff is vital to ensure accurate and timely order fulfillment.

c. Potential components or tools:

Mobile/Web Application: Developing a customer-facing mobile app or web portal for order placement and tracking.

Backend Serve

Database

Payment Gateway Integration

POS Integration

Analytics Tools: Employing analytics tools to analyze sales data, customer preferences, and other relevant metrics for business insights.

Communication Channels: Implementing real-time communication channels, such as notifications, SMS, or order display screens in the kitchen, to facilitate communication between staff members.

Security Measures