Marking Rubric

Criteria	Excellent (65-100)	Good (50-65)	Satisfactory (40-50)	Needs Improvement (30-40)	Unsatisfactory (0-30)
Understanding	Demonstrates a deep understanding of the question topic, and technical terms. Explanations are accurate, detailed, and insightful.	Demonstrates a good understanding with minor inaccuracies. Explanations are mostly detailed and clear.	Demonstrates a basic understanding with some inaccuracies. Explanations lack depth and detail.	Demonstrates limited understanding. Explanations are often unclear or incorrect.	Demonstrates little to no understanding. Explanations are mostly incorrect or missing.
Originality & Creativity	Uses highly original analogies and examples. Explanations are creative and enhance understanding.	Uses original analogies and examples. Explanations are creative but less impactful.	Uses some original analogies and examples. Explanations lack creativity and impact.	Uses few original analogies and examples. Explanations are not creative or impactful.	Uses no original analogies and examples. Explanations are not creative and do not enhance understanding.
Accuracy of Content	All information is accurate and correctly represents the concepts. No errors present.	Most information is accurate with few minor errors.	Some information is accurate but several errors present.	Information is often inaccurate with many errors.	Information is mostly inaccurate or incorrect.
Depth of Analysis	Provides comprehensive analysis of each model and term. All aspects are thoroughly explained.	Provides good analysis with minor areas needing more detail. Most aspects are well explained.	Provides basic analysis. Some aspects lack depth and thorough explanation.	Provides limited analysis with many aspects under-explaine d or not explained.	Provides minimal to no analysis. Most aspects are missing or not explained.
Use of Diagrams	Diagrams are highly relevant, well-integrated, and enhance understanding.	Diagrams are relevant and well-integrated. Properly referenced.	Diagrams are somewhat relevant but not well-	Diagrams are not very relevant and poorly integrate	Diagrams are missing or not relevant. Not referenced.

	Properly referenced.		integrated. Properly referenced.	d. Properly referenced.	
Organization & Structure	Well-organized and structured. Logical flow and clear headings/subheadings. Easy to follow.	Good organization and structure with minor issues. Mostly logical flow and clear headings/subheadings.	Basic organization and structure. Some issues with logical flow and headings/sub headings.	Poor organization and structure. Difficult to follow. Inconsistent headings/subheadings.	Very poor organization and structure. Very difficult to follow. Missing headings/subheadings.
Clarity & Conciseness	Writing is clear, concise, and free of grammatical errors. Easy to read and understand.	Writing is mostly clear and concise with few grammatical errors.	Writing is somewhat clear but with several grammatical errors.	Writing is often unclear and has many grammatical errors.	Writing is unclear and has numerous grammatical errors.
Adherence to Instructions	Fully adheres to instructions. All questions answered thoroughly within word count.	Adheres to instructions with minor deviations. All questions answered within word count.	Partially adheres to instructions. Some questions not fully answered or minor word count issues.	Limited adherence to instructions. Many questions not fully answered or significant word count issues.	Does not adhere to instructions. Most questions not answered or far outside word count.

CA1 – Programming Essentials 2

Module Title:	Programming Essentials 2
Assignment Type:	Coding project(s)
Assignment Title:	CA1
Issue Date:	TBC
Assignment Compiler:	Hugh McGovern
Weighting:	60%
Due Date:	23rd March 2025
MLO Assessed:	MIMLO1, MIMLO2, MIMLO3, MIMLO4

Introduction

- You will need to upload you python code and also your answers to the questions in section 1.
- Both project and MySQL tables and Python code needs to be submitted.
- Algorithms needs to be submitted through Moodle.
- Python source code needs to be submitted through Moodle.

- Student ID must be used as part of the file names for files submitted.

Section 1 (50 Marks)

Task Description	Marks
Explain what a constructor is and when it is used in Python programming language? (5 marks) And give an example in code of this. (5 marks)	10 Marks
When is inheritance used in OOP in Python? (6 marks) And give a code example (4 marks)	10 Marks
Define polymorphism (6 marks) and give a code example. (4 marks)	10 Marks
What is a structural pattern in Python. (6 marks) And give an example of the adaptor and decorator. (4 marks)	10 Marks
Define a name error in Python. (5 marks). Give a code example of a name error. (5 marks)	10 Marks
Total:	100%

Section 2 (50 Marks) - Dorset Bank project

Dorset College has recently been issued with a banking license from the Central Bank. In order to facilitate the operation of Dorset Bank a bespoke software application will be required to be built. Dorset Bank has requested you to develop a prototype for an internal application which their staff will be able to use to perform basic banking tasks.

Using Python and Django create a Project folder. You will need to install Django in your folder. First create a virtual environment and then install Django. You can also add the MySQL Extension for VSC. You will need to create three tables. You can use tkinter forms for the GUI.

You can then add as many Python files as required in your project. And as many classes and functions as you require. Code must be written in Python with MySQL for the tables. The basic requirements are described in the following table.

Requirement	Description
REQ1	The desktop application must be built using Python VSC with a MySQL backend. In addition you will need to create three tables in MySQL – one for account data and one for the transactions one for login details. You will need to import the MySQL connector to your new project.
REQ2	When the application is launched, a user must be presented with a log in screen similar to that outlined in Figure 1 below.
REQ3	In order to ensure appropriate use of the application only internal staff maybe allowed use this software. As a result a user must login to the application. Figure 1 shows the desired layout of the login screen. The username and password can be displayed in clear text.

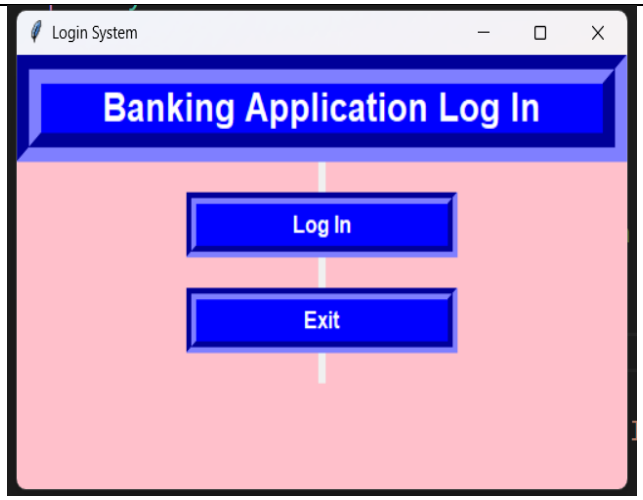
<table>
<tr><td></td><td>


Figure 1

Figure 1 is shown when a user runs the program. If the user enters an incorrect username and/or password a message should appear informing them so. You will also need to add an input mask for the password so that asterisks replace the numbers or characters of the password when typed.</td></tr>
<tr><td>REQ4</td><td>Once a user has successfully logged into the application the menu context will change. The menu should now display the following:</td></tr>
<tr><td></td><td>
<table>
<tr><td>Level 1</td><td>File</td><td>Edit</td></tr>
<tr><td>Level 2</td><td>Exit</td><td>Create Account
Delete Account
Edit Account
Transfer
Withdraw
Deposit
View Transactions</td></tr>
</table>
</td></tr>
<tr><td></td><td>Figure 2 below illustrates the updated menu.

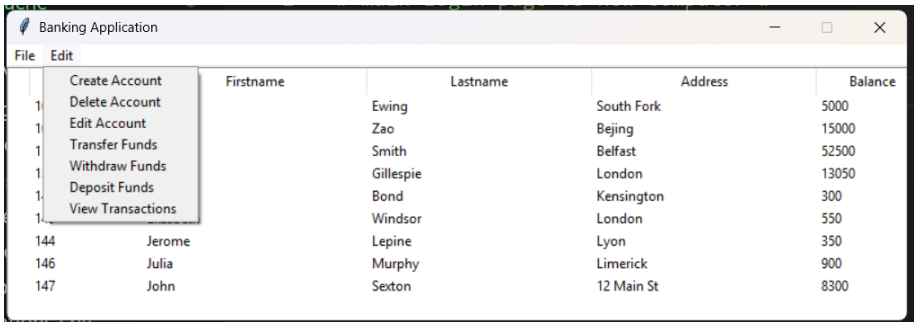


Figure 2</td></tr>
<tr><td>REQ5</td><td>In addition to updating the menu upon a successfully login a grid should be displayed containing information relating to existing bank accounts. Figure 3 illustrates an example of how this should look.</td></tr>
</table>

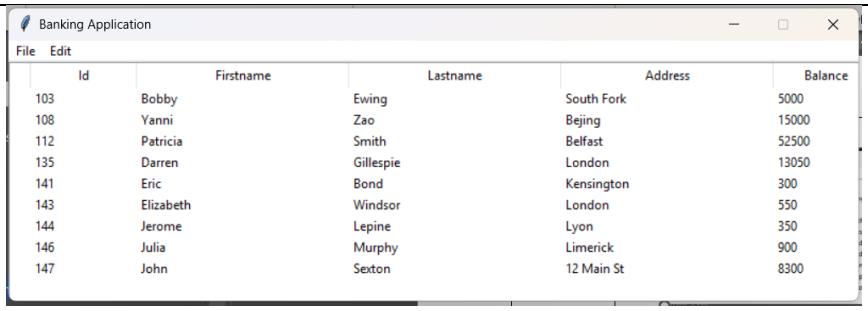


Figure 3

If no bank accounts exist then the grid should be empty. (i.e. contain no information)

REQ6

A user must be able to create a new account. To do this they select the "Create Account" menu item under "Account" from the menu bar. A screen similar to that shown in Figure 4 should be displayed.

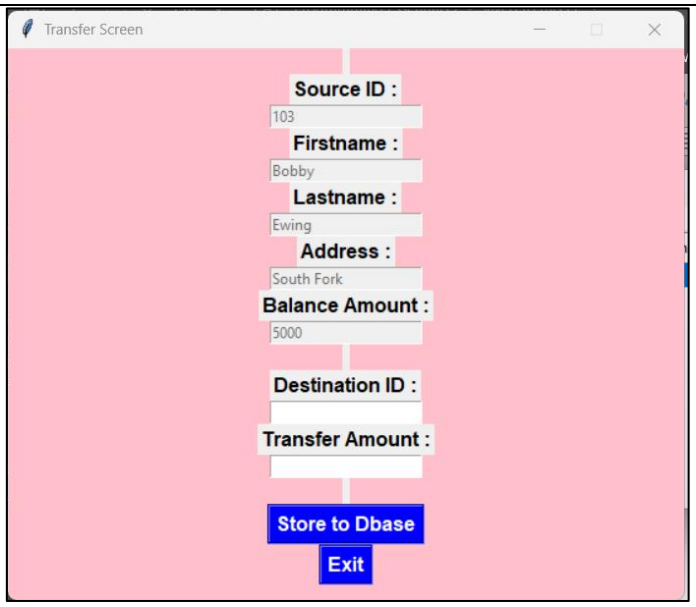
Figure 4 The following data must be captured:

	Name	Description	Data type
	First Name	The first name of the owner of the bank account	String
	Last Name	The surname of the owner of the bank account	String
	Id Number	The new account	Int (3 digits)

<table>
<tr><td></td><td></td><td>number for the account</td><td></td></tr>
<tr><td></td><td>Balance</td><td>The current balance in the account</td><td>Int</td></tr>
<tr><td></td><td colspan="3">When a new account has been successfully added to the system, the grid displaying all the accounts in the system should be updated to include the newly added account.</td></tr>
<tr><td>REQ7</td><td colspan="3">The user of the application must be able to update/edit existing accounts. The screen to do this is similar to that shown in figure 4 with the following exception:
The following fields should be read only. That is the user should not be able to modify the account balance using this screen.

• Balance.
• Id number

All other fields should be pre-populated with the relevant data for the account being updated. As per REQ6, the grid displaying all accounts should be updated to reflect any changes in the account information.</td></tr>
<tr><td>REQ8</td><td colspan="3">The user should be able to deposit and withdraw funds on behalf of the account (id) owner. A screen should be displayed where the user enters the amount of funds which has been requested to be deposited or withdrawn for a given account.
Once a deposit or withdrawal has been made, the grid displaying all the accounts should be updated to reflect the change in the account's balance.</td></tr>
<tr><td>REQ9</td><td colspan="3">The user should be able to transfer funds from one id to another. The destination id (id which funds are being transferred to).
The following data must be captured to perform this task</td></tr>
<tr><td></td><td>Name</td><td>Description</td><td>Data type</td></tr>
<tr><td></td><td>Source Id</td><td>The id of the account which the funds will be transferred from</td><td>Int (3 digits)</td></tr>
<tr><td></td><td>Destination Id</td><td>The id number of the account which the funds will be deposited</td><td>Int (3 digits)</td></tr>
<tr><td></td><td>Date/Time of transfer</td><td>A system date time when the transfer has taken place</td><td>DateTime</td></tr>
<tr><td></td><td colspan="3">It should be noted that funds must be debited from the source id.
Figure 5 illustrates the suggested layout for this screen.</td></tr>
</table>



REQ10

The system is expected to keep an audit of all funds entering and leaving an account id. Each time a deposit, withdrawal or transfer is made a transaction record should be recorded against all applicable accounts.

The balance in the transactions table needs to reflect the current balance in relevant accounts and also in the accounts table.

Figure 6 illustrates a suggested layout for the transactions associated with a particular account.

Transactions List

File Edit

TransId	Date	ReferenceID	Reference	Debit	Credit	Balance
2	2020-12-28 10:37:50	103	Transfer to 103	0	100	1700
6	2020-12-28 10:59:54	103	Transfer from 103	700	0	1000
28	2020-12-28 20:24:53	103	Deposit to 103	0	5	1005
36	2020-12-29 22:09:15	103	Transfer from 103	500	0	500
39	2021-06-14 16:49:44	103	Withdrawal from 103	100	0	400
40	2021-06-14 16:50:23	103	Deposit to 103	0	50	450
41	2021-06-14 16:53:47	103	Transfer from 103	50	0	400
45	2021-06-14 19:37:18	103	Transfer to 103	0	75	475
46	2021-10-11 20:08:14	103	Withdrawal from 103	75	0	400
51	2022-09-09 12:50:44	103	Withdrawal from 103	200	0	200

Figure 6

(a) Create four accounts of test data to test your app. **[5 marks]**
(b) Implement the project above in Python. **[45 marks]**

Notes:

- Assignment must be submitted to Moodle on or before the deadline stated.
- Failure to submit by the deadline will incur 10% penalty per day up to 5 days post submission date.
- Plagiarism software will automatically review your submission, please ensure youhave used the Harvard Referencing system throughout your submission.
- Please ensure you use the Dorset College cover page including your name, student# and CA title.