Employer Set Project - Sample Assessment Mark Scheme

General Marking Guidance

- All learners must receive the same treatment. Examiners must mark the first learner in exactly the same way as they mark the last.
- Mark schemes should be applied positively. Learners must be rewarded for what they have shown they can do rather than penalised for omissions.
- Examiners should mark according to the mark scheme not according to their perception of where the grade boundaries may lie.
- All marks on the mark scheme should be used appropriately.
- All the marks on the mark scheme are designed to be awarded. Examiners should always award full marks if deserved. Examiners should also be prepared to award zero marks if the learner's response is not rewardable according to the mark scheme.
- Where judgement is required, a mark scheme will provide the principles by which marks will be awarded.
- When examiners are in doubt regarding the application of the mark scheme to a learner's response, a senior examiner should be consulted.
- Crossed out work should be marked **unless** the learner has replaced it with an alternative response.
- Accept incorrect/phonetic spelling (as long as the term is recognisable) unless instructed otherwise.

Levels-Based Mark Scheme Guidance

Levels-based mark schemes (LBMS) have been designed to assess students' work holistically. They consist of two parts:

1) **Indicative content**
   Indicative content reflects content-related points that a student might make but is not an exhaustive list. Nor is it a model answer. Students may make some or none of the points included in the indicative content as its purpose is as a guide for the relevance and expectation of the responses. Students must be credited for any appropriate response.

2) **Levels-based descriptors**
   Each level is made up of a number of traits which when combined together articulate the quality of response that a student needs to demonstrate. The traits progress across the levels to demonstrate the different expectations of each level. When using a levels-based mark scheme, the 'best fit' approach should be used.

Applying the levels-based descriptors

Examiners should take a 'best fit' approach to determining the mark.

- Examiners should first make a holistic judgement on which level most closely matches the student's response. Students will be placed in the level that best describes their answer. Answers can display characteristics from more than one level, and where this happens markers must use any additional guidance (e.g. weighting of traits) and their professional judgement to decide which level is most appropriate.

- The mark awarded within the level will be decided based on the quality of the answer and will be modified according to how securely all traits are displayed at that level:

    - Marks will be awarded at the top of that level if the student has evidenced each of the descriptor traits securely.
    - Where the response does not securely meet all traits, the marks should be awarded based on how closely the descriptor has been met.

Task 1 - Planning a project

Indicative content and marker guidance

Gannt Chart:

- This is a large project so would expect to see tasks broken down into smaller chunks where sensible for example:
  - May show use of an Agile approach (or similar)
  - Large modules (e.g. Module Back-end database) broken down into multiple sections of development and unit testing
  - Integration testing may be split in to smaller chunks and distributed at different times when additional modules are competed
- There should be sensible use of concurrent and serial tasks for example:
  - Installing server and upgrading infrastructure could be happening while software modules are being developed
  - Integration testing would be expected after specific modules have been unit tested
  - Testing time for each module would be split up to allow testing along with other modules.
- There is no single correct way organise the plan but task orders should be sensible for example ‘create a test plan’ should occur BEFORE testing commences, acceptance testing would occur much later in the process when the system is nearer completion etc.
- The order and implementation of the project may vary significantly depending on based on the SDLC approach that they are taking (check against learner rationale). for example :
  - a RAD/agile that looks at a Minimum value product (MVP) they may 'Deploy' some of the modules very early on, after a short portion of development time, and then test and develop further, deploying more modules as they go.
  - Or they may decide that their approach is to have most modules mostly developed and then deployed later.

The rationale will show the reasoning for the chosen project development approach demonstrated in the Gantt chart points the learners may consider, although some of these will vary depending on the choices learners make in terms of organisation. These include but are not limited to:

- Staff (skills, industry expertise, experience)
  - Only one member of staff has experience in the financial industry although many have experience in the retail industry so may be familiar with risks and challenges of dealing with financial systems
  - One member of staff is very inexperienced which may be risky on such a large, and potentially risky project. More senior/experienced staff may need to take on extra work and/or provide support
  - There is no member of staff that has specific skills in designing user interfaces
- Resources
  - Learners should rationalize the choice of allocation of tasks to different members of staff
  - A justification of which server option they choose should be presented
- Time scales and costs
  - Learners should identify if the project deadline is realistic/achievable or not and provide support for any reason why the project time line is inappropriate.
  - More contingency time may needed on jobs performed by Ahad, due to his inexperience
  - Learners should discuss the cost of their solution with reference to how these costs were arrived at if the projected costs and increase in revenue make the project feasible
  - If the online sever option is chosen then the installation of the physical server task wouldn't need to be included in the plan
- Risks
  - All members of staff are new and therefore unproven in the company
  - 17 Weeks is a tight timescale for such a project and may not be feasible even slight slippage could cause the project to go over deadline
  - It is unrealistic to assume that there will be no staff absence in a 17-week period. It would be wise to build contingency time in to the project as a whole and/or the tasks assigned to each member of the team
  - Discussion of the relevant risks between physical and hosted server options and why each was selected or rejected

Assessment focus	Band 0	Band 1	Band 2	Band 3
	0	**1-2**	**3-4**	**5-6**
Gantt chart	No rewardable material	Project tasks are somewhat organised in logical and efficient manner making some use of an appropriate SDLC model to provide some accurate prediction of the project's timescales Resources have sometimes been assigned to project tasks effectively but there are some major and/or significant errors or omissions	Project tasks are organised in a mostly logical and efficient manner making use of an appropriate SDLC model to provide mostly accurate predictions of the project's timescales Resources have mostly been assigned to the project tasks effectively, but there are some minor errors or omissions.	Project tasks are organised in a thoroughly logical and efficient manner using an appropriate SDLC model to provide thoroughly accurate predictions of the project's timescales Resources have consistently been assigned to the project tasks effectively.
	0	**1**	**2-3**	**4**
Resource and cost plan		Some correct resources and accurate costs have been added to the plan resulting in an estimate of limited accuracy.	Mostly correct resources and accurate costs have been added to the plan resulting in an estimate that is largely accurate.	Fully correct resources and accurate costs have been added to the plan resulting in an accurate estimate.
	0	**1-3**	**4-6**	**7-9**
Rationale	No rewardable material	Rationale for project planning decisions demonstrate some effective consideration of: • cost, risks and benefits • order and timing of tasks • selection and allocation of resources • dependencies and prerequisites.	Rationale for project planning decisions demonstrate mostly effective consideration of: • cost, risks and benefits • order and timing of tasks • selection and allocation of resources • dependencies and prerequisites.	Rationale for project planning decisions demonstrate a thorough and perceptive consideration of: • cost, risk and benefits • order and timing of tasks • selection and allocation of resources • dependencies and prerequisites.

Task 2 - Identifying and fixing defects in an existing code

Indicative content and marker guidance

Core errors in given code. (line numbers apply to the example working code and are for guidance only expect variation based on alterations made by the learner):

- Line 9 and 10 indentation error. Should be aligned with lines 7 and 8
- Line 20 – variable **opening.investSum()** takes a user input but is not converted to a useable numerical data format
- Line 42 – incorrect name for called function stockMain() should be stocksMain()
- Line 70 & 71 – values for counter of index 'i' return the wrong year's results should be:

```
70          if i == 0 or i == 4 or i == 9:
71              print('At the end of year', str(i+1))
```

(or similar logic)

- Values are not displayed to 2.dp – should use round() or trunc(). (possible use in lines 73, 75, 77, 96, 98, 100, 136, 138, 140,142, 171,173, 175, 177
- Line 114 & 149 using the wrong value for the fees so will be outputting incorrect results (value is 0.13 but should be 0.013
- Lines 122 to 127 and 157 to 162 – the logic for the if statement is slightly wrong should be > for the if and elif not >=
- Line 164 & 165 – using wrong variable. Using **stocksMin.total** instead of **stocksMax.total**
- Functions **savingsPrint()** and **stocksPrint()** do not output the users address when giving the quote (lines 188 and 200 in example code)

The learner may have also addressed handling user error which could include addition of code to the function opening() which may include:

- Length limitation to name or phone variables
- An 'isdigit() or similar check for the phone number
- 'try..except' structure

The test plan/log should also contain inclusion of tests that show that areas of the program that appear to be coded correctly have been tested to ensure outputs are correct and the program is robust. These may include (but not limited to):

- Positive and negative values for amount invested
- Non numerical values for amountinvested
- Manual calculations of totals to check program is outputting correct values

A limited understanding of program requirements would be typified by only identifying and fixing the functional errors that would be highlighted by the IDE, but would not identify and fix logical errors.
The number of errors identified is not a hurdle between Band 2 and 3, the discriminator is the quality and appropriateness of the tests selected.

Assessment focus	Band 0	Band 1	Band 2	Band 3
	0	1-2	3--5	6-8
Use of testing to identify defects	No rewardable material	Tests selected show a basic understanding of the identified program requirements. Test log includes some appropriate test data. Testing has identified some error in the code provided.	Tests selected show a good understanding of the identified program requirements. Test log includes a good a range of normal, erroneous and extreme data. Testing has identified most errors in the code provided.	Tests selected show a thorough and detailed understanding of the identified program requirements. Test log includes a comprehensive range of normal, erroneous and extreme data. Testing has comprehensively identified the errors in the code provided.
		1	2-3	4
Understanding of the testing process		Test log shows a basic understanding of how errors/problems were identified and how they were rectified.	Test log shows a good understanding of how errors/problems were identified and how they were rectified.	Test log shows a thorough and detailed understanding of how errors/problems were identified and how they were rectified.
		1-3	4-6	7-9
The solution		Code has some functionality, but significant errors still persist. Changes made apply some precise logic and programming structures which would result in some correct outcomes.	Code is mostly functional, but some minor errors still persist. Changes made apply mostly precise logic and programming structures which would result in mostly correct outcomes.	Code is fully functional. Changes to code apply fully precise logic and programming structures throughout which would result in consistently correct outcomes

Task 3 – Designing a solution

Indicative content and marker guidance

General guidance:

- Algorithm designs should demostrate decomposition of the problem into simpler and more understood primitives.
- The design should provide high level coverage of the process as well as identify **reusable components**
- Detailed algorithms (pseudocode) do not need to be provided for ALL repeated processes. For example if the process for converting from one currency to another is very similar, the learner would not need to provide algorithms for each conversion, rather they should show how **reusable** code and may provide some additional annotation to explain the process as necessary.
- Good decomposition will show all the necessary processes and sub-processes that make up the main problem

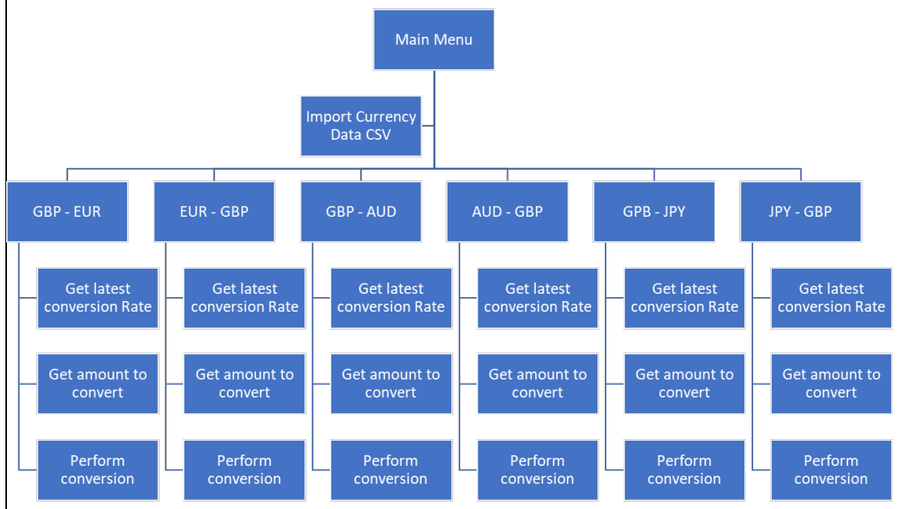
Some general characteristics of a good algorithm that may be demonstrated are:

- the steps are clearly defined
- each step is uniquely defined and should depend on the input and the result of the preceding steps
- the algorithm stops after a finite number of instructions are executed, two key constructs iterate and decide
- receives input, how much and what data is required in this case currency amount and rate of exchange
- produces appropriate type of output e.g. screen display, return value or return list, what results are required, what happens if no results can be computed maybe error
- Sensible names
- Use of key words
- Sensible indentation to show dependency
- Flow charts are allowed with correct use of symbols
- Correct structure, notation and syntax

Scenario specific characteristics may include:

- Links to CSV to get currency figures
- Sensible use of CSV or run-rime data structure to hold most up-to-date rates
- Understanding of given data such as:
    - Use of header row in CSV to identify currency conversion e.g. GBP – EUR
    - Most recent conversion rate is in the final row of the CSV
    - Date column is a string and would need converting to time/date format if it is to be used to ascertain meaningful information
    - Conversion rates are also strings and would need converting to floats
    - Conversion rates use 6 decimal places. This is good precision and looking at trends but may need rounding to fewer d.p. to be meaningful/sensible for a user.
- Options for available currency conversions
- Simplification for user e.g. choose a number form menu rather than type currency
- Handling of input – conversion from string to number and reduction of potential errors

Example high Level decomposition design



Example detailed algorithms for repeated processes:

Note – these are intended to be indicative of the types of algorithms that may be presented. These **do not** show all processes. Accept any responses that provide logically correct outcomes/solutions

Main menu

```
WHILE not_valid_input_flag = TRUE
        SEND ‘Please choose an option form the list’ TO DISPLAY
        SEND <list of options> TO DISPLAY
        RECEIVE menu_choice (float) FROM KEYBOARD
        IF menu_choice NOT float THEN:
                SEND ‘Input must be a numerical value’ TO DISPLAY
                SET not_valid_input_flag TO TRUE
        ELSE IF menu_choice <1 OR >6 THEN:
                SEND ‘That is not  a valid choice’ TO DISPLAY
                SET not_valid_input_flag TO TRUE
        ELSE:
                RETURN menu_choice
```

Get latest currency rate

```
IMPORT data handling library

READ "Task3a_data.csv"
SET dataframe TO Task3a_data.csv
SET currency TO currencies[menu_choice]
SET conversion_rate to dataframe[“currency”, -1].
RETURN conversion_rate
```

Conversion

```
 WHILE not_valid_input_flag = TRUE
        SEND ‘Please enter amount to convert’ TO DISPLAY
        RECEIVE conversion_amount (float) FROM KEYBOARD
        IF conversion_amount NOT float THEN:
                SEND ‘Input must be a numerical value’ TO DISPLAY
                SET not_valid_input_flag TO TRUE
        ELSE IF conversion_amount <1 THEN:
                SEND ‘values must be greater than 0’ TO DISPLAY
                SET not_valid_input_flag TO TRUE
        ELSE:
                RETURN conversion_amount
SET amount_recieved TO conversion_amount * conversion_rate
SEND ‘You will receive <amount_recieved>’ TO DISPLAY
```

Assessment focus	Band 0	Band 1	Band 2	Band 3
	0	1-2	3-5	6-8
Decomposition of the problem	No rewardable material	Basic or superficial decomposition of the identified problems that superficially cover the required: • inputs • processes • outputs.	Mostly detailed decomposition of the identified problems that sufficiently cover the required: • inputs • processes • outputs.	Thorough and detailed decomposition of the identified problems that comprehensively cover the required: • inputs • processes • outputs.
		1-2	3-4	5-6
Application of logical thinking and conventions		Algorithms would produce some correct outcomes as a result of: • some precise logic • some appropriate structure and sequence which is likely to be inefficient. Some use of accepted conventions.	Algorithms would produce mostly correct outcomes as a result of: • mostly precise logic • appropriate structure and sequence but which may lack efficiency. Mostly appropriate use of accepted conventions though some minor inconsistencies may still exist.	Algorithms would produce correct outcomes as a result of: • precise logic • appropriate and efficient structure and sequence. Appropriate and consistent use of accepted conventions.
		1	2	3
Communication of the design		Superficial communication of the design uses technical language which is only sometimes appropriate for the audience.	Adequate communication of the design uses technical language which is mostly appropriate for the audience.	Effective communication of the design uses technical language which is appropriate for the audience.

Task 4a - Developing the solution

Indicative content and marker guidance

The solution

- Provides developed coded solution that utilises the given code and adds the additional functionality as stated in the requirements.
- Integration of exiting code may include:
  - ‘import’ function to pull code as a whole in when needed (note given code and learner code will need to be in the same folder)
  - Integration into learner’s own code base as a function
- The solution will be well structured and modular in nature – clearly see subsections this may be separated modules or the use of Procedures, Functions or classes
- Code will be annotated to aid future maintenance of the system
- Data/information should be output in a meaningful way to the user. This may include use of a dataframe, graph and/or text based summary
- Output data should show consideration of ‘performance over time’ – Higher level responses will show greater consideration (e.g. basic level = all data for a single currency against data. Higher level = data extracted for specific time scales (e.g. last 7 days, last 14 days)
- All variables and structures will be appropriately named.
- Code should be robust typical errors that may be accounted for in this scenario include negative values, non numerical characters, entering a choice not provided in the menu

Possible areas included that contribute to ‘user experience’:

- Outputs are meaningful and make sense to the user e.g. Value outputs are accompanied by meaningful text to contextualise them
- Simplification of input processes e.g. choice of a number instead of typing currency name
- Numerical values are rounded to a sensible number of decimal places
- Use of visualization e.g. graphs for currency over time
- Helpful messages and robust input handling

Example code snippets for parts of the solution:

Note – these are intended to be indicative of the types of re that may be presented. These **do not** show all processes. Accept any responses that provide logically correct outcomes/solutions

Menu with error handling to accept user choice

```
flag = True

while flag:
    print("#############################################")
    print("###   Historical data for USD to GBP       ###")
    print("#############################################")
    print("Please choose an option form the list")
    print("1. USD to GBP last 7 days")
    print("2. USD to GBP last 14 days")
    print("3. USD to GBP last 30 days")
    print("#############################################")

    usd_choice = input("Please enter the number of your choice (1-3): ")

    try:
        int(usd_choice)
    except:
        print("Sorry, you did not enter a valid choice")
        flag = True
    else:
        if int(usd_choice) < 1 or int(usd_choice) > 3:
            print("Sorry, you did not neter a valid choice")
            flag = True
        else:
            return usd_choice
```

Integration of existing code

```
def main_menu_actions():
    if main_choice == 1:
        import Task4a_currency_conversion
    elif main_choice == 2:
        gbp_process_data()
    else:
        usd_process_data()
```

Using data structures (data frames) to extract and display a selection of data and displaying an average

```
def usd_process_data():

    main_df = pd.read_csv("Task4_data.csv")

    usd_choice = int(usd_menu())

    if usd_choice == 1:
        usd_df = main_df[["Date", "USD - GBP"]]
        seven_day_usd = usd_df.iloc[-7:]
        seven_day_avg = round(seven_day_usd["USD - GBP"].mean(),3)
        print("GBP to USD last 7 days")
        print(seven_day_usd)
        print("")
        print("The aveage convesion rate for the last 7 days is: ", seven_day_avg )
```

Use of matplotlib library to resent as a live graph:

```
seven_day_gbp.plot.line(x="Date")
```

Assessment focus	Band 0	Band 1	Band 2	Band 3	Band 4
	0	1-2	3-4	5-6	
Functionality	No rewardable material	The solution implements code with some functionality but some major errors still persist.	The solution implements mostly functional code but code may lack efficiency and some minor errors still persist.	The solution implements functional and efficient code throughout.	
		1	2	3	
Logic and programming structures		The code uses some precise logic and programming structures which would result in some correct outcomes.	The code uses mostly precise logic and programming structures which would result in sufficiently correct outcomes.	The code uses precise logic and programming structures throughout which would result in consistently correct outcomes.	
		1	2	3	
Robustness		The code handles some common user errors	The code handles most common user errors	The code thoroughly handles common, and most unexpected, user errors	
		1-2	3-4	5-6	
Security		The code mitigates against some common vulnerabilities as a result of some effective application of secure coding practices.	The code mitigates against most relevant vulnerabilities through mostly effective application of secure coding practices	The code thoroughly mitigates against relevant vulnerabilities through effective application of secure coding practices	
		1-2	3-4	5-6	7-8
Code organisation		The code is partially maintainable by a third party but would present significant difficulties through the use of: • inconsistent naming conventions • limited logical organisation • limited informative commenting	The code is partially maintainable by a third party but would present some minor difficulties through the use of: • some consistent naming conventions • some logical organisation • some informative commenting	The code is maintainable by a third party and would present only a few minor difficulties through the use of: • mostly consistent naming conventions • mostly logical organisation • mostly informative commenting	The code is easily maintainable by a third party through the use of: • consistent and appropriate naming conventions • fully logical organisation • highly informative commenting

	1-2	3-4	5-6	7-8
User experience	Basic user experience is provided through limited effective use of: • input handling • user guidance and error messages • outputs	Adequate user experience is provided through somewhat effective use of: • input handling • user guidance and error messages • outputs	Good user experience is provided through mostly effective use of: • input handling • user guidance and error messages • outputs	Excellent user experience is provided through consistently effective use of: • input handling • user guidance and error messages • outputs

Task 4b - Reflective evaluation

Indicative content and marker guidance

Indicative content will vary according to the approach learners have taken in task 4a and the effectiveness of the solution they created.

Generic features of effective evaluations are likely to include:

- the extent to which the solution meets the:
  - requirements of the set task brief
  - needs of the users.
- a justification of how the solution could be further developed/enhanced.
- specific examples from the solution to support points made
- contextualisation of any points made and explaining what they did and justifying why.

Contextulisation for this scenario may include:

- Choice of data output(e.g text, table, graph type) – which is most suitable for data over time and why
- Rounding vs truncation or reducing number of decimal places for the calculation output
- How existing code was integrated
- Choice of libraries/functions to get required data
- How most recent conversion was established and used (e.g. use of date column vs assuming data would be appended to the end and latest is always the last value)
- Use of variables (global vs local, passing data between functions)
- Input error handling – e.g. why they might have excluded text or negative values, menu options etc

Assessment focus	Band 0	Band 1	Band 2	Band 3
	0	1-2	3-4	5-6
Programming outcomes	No rewardable material	Judgements reached are somewhat supported showing a superficial or basic understanding of how well the solution met the: ● requirements of the set task brief ● needs of the users.	Judgements reached are mostly well supported showing a good understanding of how well the solution met the: ● requirements of the set task brief ● needs of the users.	Judgements reached are comprehensively well supported showing a thorough and detailed understanding of how well the solution met the: ● requirements of the set task brief ● needs of the users.
		1	2	3
Future Developments		A superficial or simplistic rationale is provided for what future developments should be implemented.	A good rationale which is reasonably well supported is provided for what future developments should be implemented.	A convincing and well-supported rationale is provided for what future developments should be implemented.