**ICT30120 – Certificate III in Information Technology**

**ICTPRG302 – Apply Introductory Programming Techniques**

**Assessment Task 2 – Practical Skills**

|  |  |
| --- | --- |
| **Student Name** | **Harrison Morris Clark** |
| **Student ID** |  |

**Assessment Overview**

This assessment provides the opportunity for you to demonstrate the following skills and knowledge:

* Design and build a simple application to specifications
* Develop algorithms
* Apply programming language syntax, sequence, selection and iteration constructs
* Test and debug a simple application
* Develop design specifications

## Instructions:

This assessment is to be completed in your own time. Time may also be made available for completing the assessment during class sessions.

You are required to enter your responses in the spaces provided in this assessment document, completed assigned activities and, follow instructions to upload resources

To achieve a ‘satisfactory’ result for this assessment you must complete all tasks and be deemed competent in all tasks by your assessor. In the event that you receive an unsatisfactory result, you will be required to review feedback from your assessor and then resubmit the assessment after making required corrections.

You will have one opportunity for resubmission. If your second assessment attempt is ‘not yet satisfactory’ you must contact your teacher or assessor to discuss how to proceed.

All responses must be your own work.

## Documents to Submit:

* This completed assessment document titled - Assessment 2 – Skills
* Mp4 screencast video of how to debug Python code
* Fully functioning Password Manager application

Upload all documents by the due date to the drop box for ICTPRG302 Assessment 2 on VU Collaborate.

## Assessment scenario – Password Manager

You work as a programmer for a small digital development agency “Apps2U”. Apps2U have asked you to develop an application on behalf of their customer “DigiCore”.   
  
DigiCore have requested a simple password manager because their staff are resorting to insecure practices to record login credentials such as writing them down on sticky notes. The password manager will be used to store and retrieve credentials for websites and other login services used by employees.

|  |
| --- |
| Task 1.1 Clarify task with required personnel Discuss the programming specification at task 1.4 with your instructor who is acting as your work supervisor. This is your opportunity to ask any questions to ensure you fully understand what the application should do. Provide brief notes or dot points of the discussion that show your attempts to clarify and understanding the programming specification. *Note that your instructor may work with you individually, in small groups or as one large group for this task* |
| *Enter brief notes or dot points of the discussion here:*  • I asked if all three credentials (URL, username, and password) need to be encrypted and was told yes  • We discussed saving data to a text file using append mode so existing entries aren’t overwritten  • I confirmed that ROT3 is fine to use for simple encryption as part of the project  • I clarified that the program should return to the main menu after each action instead of exiting |
| Task 1.2 Identify design specifications relevant to the task Complete the following design specification |
| |  | | --- | | **Design specification** | | **Overview of the project**  *This project involves creating a simple password manager that allows users to securely store and retrieve login credentials using encryption.* | | Who will use the application *Employees at DigiCore or other users who need a basic, secure way to manage and store login information.* | | **Why is the application needed***Users were storing credentials unsafely (e.g. sticky notes), which creates security risks. The application offers a safer digital alternative.* | | **What is the benefit of using the application for the client**  *It helps users store their credentials securely, improves password management habits, and reduces the chance of passwords being lost or leaked.* | | **What are the features and functions of the application? *What is it that the application does overall, what are its individual functions, how will a user navigate the application, what are the inputs, what are the outputs, are there any special features? The application allows users to:***  *• Save credentials (URL, username, password)*  *• Encrypt all stored information using ROT3*  *• View stored credentials (decrypted on display)*  *• Navigate through a simple text-based menu*  *Inputs include the URL, username, and password*  *Outputs are success messages and the display of saved credentials*  *Special features include automatic file creation and encryption for all saved data* | | **Appearance** ***How will the user interface look? How will any outputs from the program look?*** *The program uses a text-based menu in the terminal. Output is neatly formatted with labels and spacing to keep information readable.* | | **When is the application due for completion (assessment due date)**  *06/06/2025* | | **What questions do you have about the project *What do you not understand about the design of the application?***  *I wasn’t sure at first if all fields needed to be encrypted or just the password I also wanted to confirm if ROT3 was acceptable and if file creation should be automatic* | | **Summary of this design specification *Describe this entire design specification concisely*** *This is a basic password manager that securely stores login details using ROT3 encryption. Users can add and view their encrypted credentials through a simple text menu. The program is file-based and designed to improve password safety in a small business setting.* | |
| Task 1.3 Algorithm Design Produce a design of your intended program using pseudo-code or a flowchart with explanatory comments. Use the program specification checklist in task 1.4 below to guide the design of your algorithm/s  Contact your instructor for constructive and corrective feedback when complete. Make any corrections to the algorithm design and provide brief notes or bullet points of the changes.  *Note that your instructor may choose to provide feedback individually, in small groups or as a large group.* |
| *Enter your pseudo code:*  START  DISPLAY "Choose an option:"  DISPLAY "1. Save login credentials"  DISPLAY "2. View saved credentials"  DISPLAY "3. Exit"  INPUT userChoice  IF userChoice == 1 THEN  DISPLAY "Enter website/application URL:"  INPUT url  DISPLAY "Enter your username:"  INPUT userName  DISPLAY "Enter your password:"  INPUT userPassword  ENCRYPT userPassword  OPEN "credentials.txt" FOR APPENDING  WRITE url, userName, encrypted userPassword TO file  CLOSE file  DISPLAY "Credentials saved successfully."  ELSE IF userChoice == 2 THEN  IF file exists THEN  OPEN "credentials.txt" FOR READING  FOR EACH line IN file  READ url, userName, encryptedPassword FROM line  DECRYPT encryptedPassword  DISPLAY url, userName, decrypted password  END FOR  CLOSE file  ELSE  DISPLAY "No credentials found."  ELSE IF userChoice == 3 THEN  DISPLAY "Exiting program."  EXIT  ELSE  DISPLAY "Invalid choice. Please select a valid option."  END  *Enter brief notes or bullet points of any changes made after supervisor review:*  *All three fields (URL, username, password) are now encrypted before saving.*  *The WRITE step now saves encrypted URL and username, not just the password.*  *All three fields are decrypted when viewing credentials.*  *The DISPLAY step now shows decrypted values for URL, username, and password.*  *Variable names updated slightly to reflect encryption/decryption (e.g. decryptedurl, encrypted userName).* |
| Task 1.4 Create a simple application to specifications Translate your pseudo code into a Python3 script that adheres to the code layout, white space and comments recommendations of the PEP 8 Style guide for Python code. You may use the provided code examples in the appendices, as part of your script.  Use the program specification below as well as the coding checklist in the appendices to ensure that the application contains all required elements. |
| |  |  | | --- | --- | | **Program specification and checklist**  Your code will be written to the following specification | | |  | Done | | Include an options menu for the user allowing them to carry out the following actions   * Add stored credentials (username, password and URL/resource) * View stored credentials * Exit the program | **yes** | | Return to the menu after each action has completed | **yes** | | Create a text file for credential storage if a text file does not already exist | **yes** | | Append new records to the text file without overwriting previous entries | **yes** | | Display the text file contents in a visually presentable way including spacing and headings | **yes** | | Handle any input from the user and carry out actions, without errors | **yes** | | Include embedded explanatory comments (#) to clarify the meaning of the code | **yes** | | Provide simple rot3 encryption on all written data and, decryption on read data | **yes** | |
| Task 1.5 Debug Debug your application   * Use an IDE and its inbuilt debugger to debug your script * Using a screen casting tool of your choice, create a short video (3 minutes maximum) with commentary that shows the IDE debugger in action, including the variables contents changing, stopping at a breakpoint, stepping over a function, stepping into a function and, identification of the cause of an error. *Be brief by not including any more content than is required for the task* * Upload a copy of your video to the assessment drop box for this unit. |
| 1. *Provide a list of three semantic errors you have encountered and how you rectified them.  Note that syntax errors are NOT acceptable. The errors MUST be caused through incorrect logic*  |  |  | | --- | --- | | **Error** | **Rectification** | | *The password was encrypted, but the username and URL were not* | *Updated save\_credentials() to encrypt all three fields before saving* | | *Program displayed encrypted data instead of decrypted* | Updated view\_credentials() to decrypt each field before displaying | | *I found a logic error that caused extra blank lines between each row of saved credentials. The program still ran, but the output didn’t look right.* | I removed the \n at the beginning of the print line inside the view\_credentials() function. This fixed the formatting and the credentials now display in a clean table. | |
| Task 1.6 Test Develop test cases to confirm the code meets the program specifications. Record the Test cases below. |
| |  |  |  | | --- | --- | --- | | **Description:  *(what is being tested)*** | **Expected response or output** | **OK** | | *e.g. Add a new record for the first time* | *Text file is created; username, password and URL are add to the text file. Password is encoded.* | *Yes* | | *Save valid credentials* | data is encrypted and saved | yes |  |  |  |  |  | | --- | --- | --- | --- | | |  | | --- | | View credentials | | Data is decrypted and displayed | yes | | File doesn’t exist | File is created automatically | yes | | |  | | --- | | Invalid menu input | | |  | | --- | | Shows error message, doesn't crash | | yes | | |  | | --- | | Special characters | | Handled without error | yes | | Handled without error | |  | | --- | | Program exits cleanly | | yes | |  |  |  | |
| Task 1.7 Gain feedback, review and finalise Seek feedback from your supervisor and review the code.   * Contact your instructor, acting as your supervisor, when your application is complete to confirm that your application meets the initial design specifications and program specifications. Document this discussion via bullet points or brief notes, make any required adjustments to the code and, obtain final verbal sign-off. * Save and upload the final version of your application to the drop box for this assessment. |
| *needed to change it as a record row, website, username and password.* |
| **Appendix A** – Coding Checklist |
| |  |  | | --- | --- | | **Coding Checklist**  Your code must include following criteria. *The checklist is given to* ***ensure*** *all aspects of the assessment are covered. Your assessor will confirm that your code includes the criteria when marking your submission* | | | Criteria | Tick when complete | | At least one of each of the following:   * Global variable * Local variable | **yes** | | At least two library functions  *(internal or external)* | **yes** | | At least one self-created function | **yes** | | Clarifying comments | **yes** | | A data structure  (i.e. list, dictionary, tuple or set) | **yes** | | Manipulation of strings | **yes** | |
| **Appendix B – Provided code**  You may use the following premade code in your script  **Menu system**  # Give the user some context.  print("\nThis program…………………………")  # Set an initial value for choice other than the value for 'quit'.  choice = ''  # Start a loop that runs until the user enters the value for 'quit'.  while choice != 'q':      # Give all the choices in a series of print statements.      print("\n[1] Enter 1 to create an encryption key.")      print("[2] Enter 2 to …….")      print("[3] Enter 3 to……")      print("[q] Enter q to quit.")        # Ask for the user's choice.      choice = input("\nMake your choice ")        # Respond to the user's choice.      if choice == '1':          print("\nEnter a name for the encryption key\n")      elif choice == '2':          print("\nEnter …….\n")      elif choice == '3':          print("\nEnter ……\n")      elif choice == 'q':          print("\nExiting the menu\n")      else:          print("\nInvalid option, please try again.\n")    # Print a message that we are all finished.  print("Program exit.")  **ROT3 encryption**  clearText = "myPassword"  charSet="0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz`~!@#$%^&\*()\_-=|\}]{[\"':;?/>.<, "  encText = "".join([charSet[(charSet.find(c)+3)%95] for c in clearText])  print(encText) |