

**ANL252 (Online)**

**Python for Data Analytics**

# **Tutor-Marked Assignment**

**July 2023 Presentation**

**Submitted by:**

|  |  |
| --- | --- |
| **Name** | **PI No.** |
| **Poh Sze Ying** | **M2170235** |

**Question 1a**

Why plagiarism in coding happens.

* People share their codes on code-sharing sites and forums such as Stack Overflow and GitHub (Lee, 2021).
* Most programming codes are similar.
* Borrowing of codes from online platforms (Ondich, 2023).
* Not able to debug our code.
* Not able to solve the programme (Lee, 2021).

How to avoid this issue

* Attempt to solve the code without using the internet first (Ondich, 2023).
* Include a succinct justification and citation of the source code (Ondich, 2023).
* Have integrity over the source codes used in their assignment (Lee, 2021).
* Not to mix code from the web with our own unless properly cited (Bailey, 2017).
* Create unique variable names which can be understood (Bailey, 2017).

**Question 1b**

|  |
| --- |
| citz\_info = {}  username = str(input("Please enter your name:"))  userage = int(input("Please enter your age:"))  usercitizen = str(input("Are you a Singaporean Citizen (Y/N):"))  citz\_info["Name:"] = username  citz\_info["Age: "] = userage  citz\_info["Citizenship: "] = usercitizen  print(citz\_info)  b = citz\_info.get("Age: ")  c = citz\_info.get("Citzenship: ")  if c == "Y":  if b >= 21:  print("You are eligible to vote.")  else:  print("You are not eligible to vote")  else:  print("You are not eligible to vote") |

This Python program will determine if the user is eligible to cast a vote. The program would prompt the users to enter information such as their name, age, and citizenship before storing it in a dictionary. The user-inputted data would then be printed out.

Next, I would create the variables b and c to extract the age and citizenship respectively, from the dictionary. To determine if the user is entitled to vote, the program would first check the user's citizenship. If the user satisfies the first criteria, the program then checks the user's age in the dictionary to determine whether they are of voting age.

**Question 1c**

|  |
| --- |
| username = str(input("Please enter your name:"))  userage = int(input("Please enter your age:"))  usercitizen = str(input("Are you a Singaporean Citizen (Y/N):"))  citz\_info = [username, userage, usercitizen]  print(citz\_info)  if citz\_info[2] == "Y":  if citz\_info[1] >= 21:  print("You are eligible to vote.")  else:  print("You are not eligible to vote")  else:  print("You are not eligible to vote") |

Output:

|  |
| --- |
| Please enter your name: Andrew  Please enter your age:21  Are you a Singaporean Citizen (Y/N):Y  ['Andrew', 21, 'Y']  You are eligible to vote. |

* Readability of the code in a concise manner.
* Ease of accessing information in the list using an index instead of creating new sets of codes to access information.
* Instead of creating three sets of code to input the information keyed by the users, we could use one code to store the information in the list.

**Question 2**

In terms of readability, the codes are readable as they are typed in a sequence of each line being executed.

The reliability of the code refers to how the code can run without failure (Bellairs, 2019). When program was running the code for the price of an item, there was no indication if the inputs were in strings or integers, so the program would still run even if we were to give a string input. I would change the code into price\_of\_item = float(input("How much is it (in SGD)?")) instead of price\_of\_item = input("How much is it (in SGD)?")).

Another issue encountered was the program would stop when users entered the incorrect products, rather than prompting them to re-enter the correct product. Instead of using break, I would change to continue, where the program would prompt the users to input the correct item names and the program would only stop when the user inputs “no”.

Improved code as shown below:

|  |
| --- |
| products = ['laptop','mouse','webcam','keyboard','speaker']  query = 'yes'  updated\_items = []  print(f'We have a list of products here: {products}.')  while query == 'yes':  item = str(input("Hello! What do you want to buy?"))  if item not in products:  print(f'Wrong product! Please try again.')  continue    price\_of\_item = float(input(f'How much is it (in SGD)?'))  #check if the input is in float  if type(price\_of\_item) != float:  print(f'Wrong input type! Please try again.')  continue    entered\_input = [item, price\_of\_item]  updated\_items.append(entered\_input)  query = str(input("Would you like to continue? (yes/no)"))    print(f'This is our updated shopping list: {updated\_items}') |

**Reference list**

1. Lee, C. (2021, July 14). What is source code plagiarism and what does it have to do with academic integrity? *Turnitin*. Retrieved August 25, 2023, from <https://www.turnitin.com/blog/what-is-source-code-plagiarism-and-what-does-it-have-to-do-with-academic-integrity#:~:text=Source%20code%20plagiarism%E2%80%94otherwise%20known,as%20your%20own%20without%20attribution>.
2. Ondich. (2023, March 6). *Plagiarism and writing code - Carleton College*. Writing Across the Curriculum. Retrieved August 25, 2023, from <https://www.carleton.edu/writing/plagiarism/citing-code/>
3. W3 Schools. (n.d.). *Python - Add Dictionary Items*. Retrieved August 27, 2023, from <https://www.w3schools.com/python/python_dictionaries_add.asp>
4. W3 Schools. (n.d.). *Python – Nested If*. Retrieved August 27, 2023, from <https://www.w3schools.com/python/gloss_python_if_nested.asp>
5. W3 Schools. (n.d.). *Python - Access dictionary items*. Retrieved August 27, 2023, from <https://www.w3schools.com/python/python_dictionaries_access.asp>
6. W3 Schools. (n.d.-c). *Python continue Keyword*. Retrieved September 1, 2023, from <https://www.w3schools.com/python/ref_keyword_continue.asp>
7. Bailey, J. (2017, November 8). Plagiarism and Programming: How to code Without Plagiarizing. *Turnitin*. Retrieved August 27, 2023, from <https://www.turnitin.com/blog/plagiarism-and-programming-how-to-code-without-plagiarizing-2>
8. Bellairs, R. (2019, July 11). *What is code quality? Overview + How to improve code quality*. Perforce Software. Retrieved September 1, 2023, from <https://www.perforce.com/blog/sca/what-code-quality-overview-how-improve-code-quality#:~:text=Maintainability%20measures%20how%20easily%20software,single%20metric%20to%20ensure%20maintainability>.