**ANL 252**

**LIM SOON MENG**

**Y2310177**

**15/09/2023**

**Question 1**

Part A

Plagiarism in coding can happen due to the following reasons:

* General plagiarism where the offender fails to properly attribute and credit the original author for their work and idea.
* The offender may include publicly published codes then are widely adopted and neglect to properly cite them. (Bailey, 2017)
* The offender may fail to properly credit all members that contributed in collaborative projects. (Bailey, 2017)
* Original work may be misunderstood as copied code if similar code was previously published. (Heard, 2019)

Ways to avoid coding plagiarism:

* Properly cite codes that are reproduced in our work and credit the authors accordingly for their work and ideas.
* Document how we derive our work through comments in the code and accompanying documentation. (Heard, 2019)
* Attribute work done to corresponding members in a collaborative project.
* Check online for publicly published codes that are similar to our own work before submitting.
* Avoid using other’s code unless necessary. (Heard, 2019)
* Provide citations within the code itself.

Part B

Script to simulate rolling of dice

"""

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Title: Python Game : Rolling The Dice

\* Author: PFB Staff Writer

\* Date: June 11, 2023

\* Availability: https://www.pythonforbeginners.com/code-snippets-source-code/game-rolling-the-dice

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

"""

import random

min\_value=1

max\_value=6

roll\_again = "yes"

while roll\_again == "yes" or roll\_again == "y":

print("Rolling the dices...")

print("The values are....")

value1=random.randint(min\_value, max\_value)

value2=random.randint(min\_value, max\_value)

print(value1,value2)

roll\_again = input("Press 'y' or 'yes' to roll the dices again.")

print("Have a good day.")

The above script simulates the rolling of two 6-sided dice and prints the output value.

The first line of code imports the random module. ‘min\_value’ and ‘max\_value’ is first specified to ensure that the values returned are between 1 and 6. Two results are then randomly generated using the randint() method and printed. The while loop is used in conjunction with the variable roll\_again to allow users to quickly generate a new pair of values.

Part C

Modified Script

"""

Modification of a die-rolling script for the purpose of a Dungeons & Dragons game.

Original script is as follows:

\* Title: Python Game : Rolling The Dice

\* Author: PFB Staff Writer

\* Date: June 11, 2023

\* Availability: https://www.pythonforbeginners.com/code-snippets-source-code/game-rolling-the-dice

"""

import random

new\_dice = "yes"

#while loop to allow users to change their dice

while new\_dice == "yes" or new\_dice == "y":

die\_sides=int(input('How many sides do the dice have?'))

die\_count=int(input('How many dice are there?'))

min\_value=1 #min number of the die roll

max\_value=die\_sides #max value of the die roll is the nubmer of sides

results=[] #empty list to store values

roll\_again = "yes"

while roll\_again == "yes" or roll\_again == "y":

print("Rolling the dices...")

print("The values are....")

#for loop to generate values for the number of specified die and add values to results list

for n in range(0, die\_count):

value=random.randint(min\_value, max\_value)

results.append(value)

print(f'The values are {results}.')

results=[] #reset list to empty for next die roll

roll\_again = input("Press 'y' or 'yes' to roll the dices again.")

new\_dice = input("Press 'y' or 'yes' to change your dice.")

print("Have a good day.")

Output

How many sides do the dice have?10

How many dice are there?10

Rolling the dices...

The values are....

The values are [4, 6, 4, 8, 9, 1, 6, 7, 8, 10].

Press 'y' or 'yes' to roll the dices again.y

Rolling the dices...

The values are....

The values are [10, 8, 1, 4, 6, 3, 4, 5, 3, 8].

Press 'y' or 'yes' to roll the dices again.n

Press 'y' or 'yes' to change your dice.y

How many sides do the dice have?20

How many dice are there?5

Rolling the dices...

The values are....

The values are [7, 3, 9, 3, 20].

Press 'y' or 'yes' to roll the dices again.n

Press 'y' or 'yes' to change your dice.n

Have a good day.

The use case of my modified script was assumed to be for a Dungeons and Dragons tabletop game. Modifications were made for the following reason:

* The number of sides of dice used during the game is undetermined and can change throughout the game.
* The number of dice used is undetermined and can change throughout the game.
* Script needs to allow players to change their die sides and die counts whenever they wish.

Modifications made above were to add in new features not in the original script to better fit my use case scenario. Unused code or redundant code can also be removed in the use case to make the script more efficient.

**Question 2**

Improved code for code provided in Appendix 1

products\_database = {'laptop':2000, 'mouse':80, 'webcam':150, 'keyboard':180, 'speaker':450}

products = list(products\_database.keys())

count = 0

total = 0

query = 'yes'

updated\_items = []

total = 0

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.')

break

try:

count = int(input(f'How many {item} would you like to buy?'))

except ValueError:

print('Please key in a number.')

break

subtotal = products\_database[item] \* count

total = total + subtotal

new\_entry = [f'{count} number of {item} for ${subtotal}']

updated\_items.append(new\_entry)

print(f'You have selected:\n{updated\_items}\nFor a total of ${total}.')

query = str(input("would you like to continue? (yes/no)"))

print(f'This is our updated shopping list.\nYou have selected:\n{updated\_items}\nFor a total of ${total}.')

I interpret the original code as a script for customers visiting a store to enter a list of products they wish to buy and have it generate a list of items with the cost as output. The following changes have been made to the code provided in Appendix 1.

First, the ‘products’ list has been changed to a dictionary with the products name as its key and the cost of each item as the value. This is to avoid having clients key in their own prices as done in the original script. The prevents erroneous and exploitative entries by customers and instead relies on the values stored in the dictionary list. This makes the system more reliable. The store can modify the values stored in the dictionary in the future as required.

Next, line 17 to 21 is added as a way for customers to input the number of each item they wish to purchase. This gives consideration that the customer may wish to purchase more than one copy on the item and allows the script to capture this. This is done in a try-except clause to return error messages if anything other than an integer in keyed in.

Line 22 to 26 does a quick multiplication using the number of items to be purchased provided previously by customers against the pre-stored price value in the products\_database dictionary. A string is then printed to reflect to customer how many of each item they have selected, the cost for each item selected and the total cost for all items. This allows the customers to quickly understand at a glance what items have been selected thus far and their costs.

*References*

Bailey, J. (2017, November 8). *Plagiarism and programming: How to code without plagiarizing*. Turnitin. https://www.turnitin.com/blog/plagiarism-and-programming-how-to-code-without-plagiarizing-2

Heard, L. (2019, December 19). *Plagiarism and programming: How to code without plagiarizing: It briefcase*. IT Briefcase - IT Briefcase is a targeted online publication that attracts qualified business and IT professionals who are actively researching business integration solutions. Some of the topics we cover include BI, BPM, Cloud Computing, Data Storage, Health IT and Open Source. A full list of the topics we cover can be found on the right hand side of our website. https://www.itbriefcase.net/how-to-code-without-plagiarizing

University of Arkansas. (n.d.). *Research guides: Computer science & computer engineering: Citing programming code*. Citing Programming Code - Computer Science & Computer Engineering - Research Guides at University of Arkansas. https://uark.libguides.com/CSCE/CitingCode