MASTER OF SCIENCE IN
DATA SCIENCE AND ANALYTICS

HW 1: Supermarket Game

DSA 8101: Fundamental Computing Concepts

HW deadline 8th July 2022.

This assignment deals with the following topics:
- Getting user input
- Error checking
- Variables & data types
- Conditionals

General Idea of the Assignment

In this assignment, you will implement a supermarket shopping "game".

Imagine you are shopping in a supermarket. This supermarket sells only four items: lottery tickets, bread, milk, and soda. Lottery tickets cost KES 200 each, bread costs KES 95 each, a packet of milk costs KES 70 each, and sodas cost 60 each.

You have KES 400 to start with. At the beginning of your shopping trip, you are offered the opportunity to buy a lottery ticket for a chance to win KES 200 – KES1,000. You are then sequentially asked whether you want to buy bread, milk, and sodas. If you choose to purchase an item, you are asked to specify the quantity of this product. At the end of your shopping trip, you are provided with a list of products purchased and how much money you have left.

Provide program implementation:

Create a supermarket.py file which includes:
1. 4 defined variables storing the unit price of an individual lottery ticket, the unit price of an individual loaf of bread, the unit price of an individual packet of milk, and the unit price of an individual soda.
2. 2 defined variables storing the initial money the user has and the money the user has spent.

3. 4 defined variables storing the amounts of lottery tickets, bread, milk, and sodas the user has purchased.

Program logic:
1. The user will be given KES 400 to start shopping.
2. First, Ask the user their name then print a welcome message along with a list of products and their unit prices.
3. Next, tell the user how much money they have available and ask if they want to purchase a lottery ticket.
   a. If the user inputs "y" or "Y", process a lottery ticket purchase (see Step 3b). If the user inputs anything else (e.g. "n" or "N"), print a message saying that no lottery tickets were purchased and move on to the next item.
   b. If the user chooses to purchase a lottery ticket, you will need to use the random module. The probability that the user wins the lottery is 33%. You can generate a random int from 0-2 to simulate this probability.
      i. If the user loses, print a message informing them that they did not win the lottery, and move on to the next item. Remember to deduct the KES 200 the user spent on the lottery ticket from their available money and increase the amount of lottery tickets purchased by 1.
      ii. If the user wins, you then need to calculate their winnings. You can use winnings = random.randint(2, 10) to generate a random int from 2 – 10 and store it in a variable "winnings". Consider this money earned and add it back to the money the user has available. Remember to also deduct the KES 200 the user spent on the lottery ticket and increase the amount of lottery tickets purchased by 1. Finally, print a congratulatory message to the user telling them the value of the lottery ticket.
4. Next, tell the user how much money they have available and ask if they want to purchase bread.
   a. If the user inputs "y" or "Y", process a bread purchase (see Step 4b). If the user inputs anything else (e.g. "n" or "N"), print a message saying that no bread was purchased and move on to the next item.
   b. If the user chooses to purchase bread(s), ask them how many loafs they want to buy.
      i. You should cast the input to an integer and catch the error if the input cannot be cast to an integer. In this case, ignore the input, print a friendly message reminding the user that only numerical values are accepted and move on to the next item.
      ii. If the user entered valid integer input, calculate the money they will need to pay. To do this, use the unit price of the item and the desired amount of the item. Then print the amount the user wants to purchase and how much it will cost. For example, "The user wants to buy 1 loaf. This will cost Ksh 55." If the user doesn't have enough money to pay, print "Not enough money" and move on to the next item. If the user has enough money, add the number of loafs(s) purchased to the variable representing the total amount of apples that the user has purchased and decrease the money that the user has left.

5. Repeat Step 4 for milk and for sodas.
6. At the end of the shopping trip, tell the user how much money they have left and print the following information as well: number of lottery tickets purchased, amount of lottery winnings. (this should be KES 0 if the user did not purchase a ticket or did not win the lottery), number of loaves(s) purchased, number of milk packet(s) purchased, number of soda(s) purchased.

Program Output

Print out what the program is doing as it goes along. Store the output in a template_behavior_{user_name}.txt file which shows runs of the program.

Submission

Your submission should include:
- supermarket.py - the source code for your game This file must include a header, in the form of a multi-line comment at the top of your script, that contains (each on its own line):
- Your name
- Your Strathmore University ID
- Statement of work. Either:
  o A list of resources you used and/or people you received help from (including TAs/Instructor)
  o A statement that you worked alone without help

Bonus Points/Advanced

1. Handle unexpected user inputs
2. Thanks to a BlueBand promotion add the chance (15%) to win KES 50 – KES 200 if the user buys at least 2 loaves of bread and a packet of milk.
3. Add the chance to win 50% discount on the entire shopping at the end if the user guesses correctly the number the attendant would think of between 0 and 20. Three guesses max. After each guess the attendance should give a hint as to whether the correct answer is higher/lower that the guessed number.

Evaluation

Correctness - 20 pts

Does the game work as expected? Did you follow the directions exactly? Is your math correct? If the user doesn't have enough money or the user enters invalid input, does the program work well (and print a useful message)? Do you print welcome/goodbye messages with clear information for the user?

Comments – 5 pts

Did you add clear and descriptive comments to all non-trivial lines of code?

Bonus – 10 pts