Workshop - 5

<u>Workshop Value</u>: 10 marks (5% of your final grade)

<u>Learning Outcomes</u>

Upon successful completion of this workshop, you will have demonstrated the abilities:
- to decipher and identify a problem
- to analyze and decompose a problem
- to identify the required detailed steps to solve a problem
- to communicate the solution to fellow peers and non-technical business persons

<u>Workshop Grading and Promotion Policy</u>

Workshops for this course will be assessed using the following criteria:
- Workshops must be completed before the class time to be graded
- You must successfully complete 8 workshops (if more than 8 are completed, the best 8 will be used)
- Each student is expected to be a presenter of the workshop solution at least once by the end of the term
- Workshop solutions and presentations will be evaluated using the published workshop rubrics

---

Workshop Overview

Black boxes are widely used and referred to in the world of programming. They represent specific processes that are hidden from the consumer (end-user, programmer, or conceptually by management etc.). In the programming sense, this can be a large-scale process (a system or application) or be as minute as a function (such as a shared Application Program Interface [API], or a web-service).

In the case of "functions", each function should be limited to a very specific task. Although not necessary, most of the time functions will require some input information and use that input to produce some meaningful output (or result). This workshop is focused on function-based black boxes that do a very specific task, and using those defined functions from a main overall process.

Workshop Details

A customer has several products in a shopping cart from a grocery store (perishable and non-perishable products). The customer needs to pay for the items using the "self-checkout" option.

There are three (3) main **sub-processes** (black boxes) that occur:

1. Add **non-perishable** item to the order (has a SKU barcode)
2. Add **perishable** item to the order (<u>no</u> SKU barcode)
3. Payment

The entire checkout process needs to be defined. The main process should refer to the sub-processes (functions/black boxes) when and where possible.

Your Tasks

Using the "Computational Thinking" approach to problem solving:

1. **[Logic 1]** Define the <u>main overall process</u> required using a "self-checkout" option at a grocery store.

2. Define the specific functions (black box) for each of the sub-processes. Be sure to determine what inputs might be required (if any) for it to do its task:
   - **[Logic 2]** Adding item to order: **Non-perishable**
   - **[Logic 3]** Adding item to order: **Perishable**
   - **Payment**

3. Refine the main overall process (from #1) so it refers to the defined sub-processes (black boxes) where applicable (this should result in simplifying the main overall defined process).

4. In total, you should have four (4) defined sections of pseudo code (with complimentary flowcharts for each) that illustrate and communicate the defined entire solution.

5. Presenters will create a video.

Task	Subtask	Member(s)	Marks	Comments
Pseudocode	Logic 1	1	40%	
	Logic 2	2	40%	
	Logic 3	3	40%	
	Combined	1-3	60%	
FlowChart	Logic 1	4	40%	
	Logic 2	5	40%	
	Logic 3	6	40%	
	Combined	4-6	60%	
Video	Presentation	2 or 5	100%	Members rotate weekly