Part 2 - Program Design

Contents

[1. Heading 1 3](#_Toc82166771)

[1.1 Heading 2 List 3](#_Toc82166772)

1. Psuedocode for Requiremnt 2
   1. Initial Psuedocode

The code below if from the previous development team who have since resigned from the project. The purpose of this is to adjust and improve the logic of this code.

*Get the number of pizzas to order from user (input)*

*Fine*

*If number of pizzas less than or equal to zero then*

*Validation error*

*Else*

*Divide the number of pizzas by 2 and check the remainder*

*Check what option is best using the division by 2*

*Needs to be put after the ‘End if’*

*Print the best buy option and the amount payable*

*Needs to be expanded*

*Check what option is best using the division with 3*

*Divide the number of pizzas by 3 and check the remainder*

*End if*

This code does not account for the amount of pizzas being 1.

* 1. Improved Pseudocode

You use **divide** and **modulus** operators to get how many sets of **three** pizzas and find the remainder respectively. For divide to work as intended, the value of the amount of Pizza must be explicitly cast as an int data type (ints round down to whole numbers).

We can remove if conditionals to initially check for one or two pizzas as now they will be handled by the modulus operation.

*Get the number of pizzas to order from user (input)*

*Add a totalCost variable to store the price*

*If number of pizzas less than or equal to zero then*

*Validation error*

*Else*

*Divide the* ***number of pizzas*** *inputted by 3* ***(CAST THIS VALUE TO AN INT)***

*Then multiply this value by 14, add the result to totalCost*

*If modulus of the number of pizzas equals 2*

*Add 12 to totalCost*

*Else if modulus of the number of equals to 1*

*Add 6.45 to totalCost*

*End if*

*Return the totalCost*

*TotalCost will then be passed to an InnerHTML element on the .html*

This solution works because of the ParseInt(). And the Modulus work hand in hand to return the correct value. For example:

* If the user inputs 2 pizzas,
  + The divide will return 0 Calc. 0 × 14 = 0
  + The modulus returns 2 Calc. +12 = 12

**Cost** **=** **12**

* If the user inputs 3 pizzas,
  + The divide will return 1 Calc. 1 × 14 = 14
  + The modulus returns 0 *Wont trigger*

**Cost =** **14**

* If the user inputs 13 pizzas,
  + The divide will return 3 Calc. 3 × 14 = 42
  + The modulus returns 1 Calc. + 6.45 = 48.45

**Cost =** **48.45**

For more clarity and safety, we can assign the values of the divisions and modulus operators to their own variables – **groupsOfThree**, **oneOrTwo**. We can then add them to the totalCost variable.