Assignment P1

Due Date: February 18

Purpose
Yippee! It’s time for you to try out some basic Python constructs in a short, but exciting, program. More fun than humans should be allowed to have!

Problem
The Rinky Dinky Dooflingie Company of Walla Walla Washington wishes to send out its dooflingies. To do this, the company has four sizes of boxes: huge, large, medium, and small which hold 30, 10, 5, and 1 dooflingie, respectively. Rinky Dinky is very picky about things, so boxes to be shipped must be FULL, and only the MINIMUM number of boxes should be used. Finally, the cost of the packaging should be computed. The cost of a huge box is $2.75, large is $1.50, medium is $1.00, and small is $0.50.

Input
The program should prompt the user for a number of dooflingies, an integer. You may assume the user will input a non-negative value.

Output
The output should display the total number of dooflingies (this is separate from your prompt and the user input!) followed by an aligned table showing all the box sizes and the number of each size needed, as well as the total box cost incurred. At the bottom of the table, the total number of boxes and the total box cost should be shown. All values must be right justified, with decimals aligned. All dollar values should be shown with two decimal places. There should be a dollar sign before the final total.

For example, if 148 dooflingies were entered, the results may look like the following (your output format does not have to look exactly like this!):

Number of dooflingies: 148

<table>
<thead>
<tr>
<th>Boxes</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Huge boxes:</td>
<td>4</td>
</tr>
<tr>
<td>Large boxes:</td>
<td>2</td>
</tr>
<tr>
<td>Medium boxes:</td>
<td>1</td>
</tr>
<tr>
<td>Small boxes:</td>
<td>3</td>
</tr>
<tr>
<td>Totals:</td>
<td>10</td>
</tr>
</tbody>
</table>

If there are 0 boxes of a specific size needed, it’s fine to display 0 boxes.

Specifics

- Use named global variables for the box sizes and costs.
- Include a good introductory comment, as described in class.
- Comment all variables.
• Use good (mnemonic) identifiers.

• The program should be properly indented.

• Maximum points will be given for the problem being solved using code that is as efficient as possible.

Notes

• You do not need if statements or loops in any part of this program.

• The name of your source code file (the one that ends with .py) should be your first initial, last name and project name; e.g., mgousieP1.py.

• To turn in the project, send your source code as an attachment to mgousie@wheatoncollege.edu. Submit by 11:59:59 on the due date for the project to be on time.

• A printed version of your source code is due in class on February 19th. Write/print and sign the Wheaton Honor Code Pledge on what you turn in: “I have abided by the Wheaton College Honor Code in this work.”

• Save all of your work until your project is returned.

• Before turning in your project, be sure to test with the Wing IDE if you are using another environment for development.

It’s almost too easy.
– Dr. Schulz in my Calculus III class