



Republic of the Philippines  
**CAMARINES SUR POLYTECHNIC COLLEGES**  
Nabua, Camarines Sur



**COLLEGE *of* COMPUTER STUDIES**

---

**Final Project and Final Exam**  
Defensive Programming  
CSEC-311

Modification of Python-based  
projects with Defensive  
Programming

Submitted by:

Justine Labios  
Mae Julienn Mata  
Marc Chrsitian Tumaneng



## **COLLEGE *of* COMPUTER STUDIES**

---

### **I. Project Description**

The Weight and Height Converter Application is a Python-based graphical user interface (GUI) program that enables users to easily convert weights and heights between different units. The application is an evolution from a basic weight converter to a more sophisticated tool that incorporates Object-Oriented Programming (OOP) principles, modularization, enhanced functionalities, and extended capabilities to handle both weight and height conversions, providing users with a versatile and user-friendly tool for converting weights and heights. The application's modular and organized codebase, along with its enhanced features, makes it a valuable resource for individuals needing quick and accurate unit conversions.

### **II. What did you do? Original vs. Modified (code)**

The changes from the original to the modified code involve a shift towards OOP, modularization, enhanced styling, improved input validation, and the addition of new features. The original code is a weight converter using the Tkinter library in Python. It exhibits a procedural structure, with all functionalities implemented in a linear sequence. The only purpose of the code is to convert a given weight in kilograms into grams, pounds, and ounces. The modified code on the other hand represents a substantial evolution from the original code, transforming a basic weight converter into a more sophisticated and user-friendly , height and weight converter. This iteration incorporates Object-Oriented Programming (OOP) principles, modularization, improved styling, and enhanced functionalities. We add new functionalities such as displayed instruction, height conversion, clear button and viewing recent conversion results.



### III. Screenshots of the Original vs Modified

#### Original Code

```
1 # Python program to create a simple GUI
2 # weight converter using Tkinter
3
4
5 from tkinter import *
6
7
8 # Create a GUI window
9 window = Tk()
10
11 # Function to convert weight
12 # given in kg to grams, pounds
13 # and ounces
14 def from_kg():
15
16     # convert kg to gram
17     gram = float(e2_value.get())*1000
18
19     # convert kg to pound
20     pound = float(e2_value.get())*2.20462
21
22     # convert kg to ounce
23     ounce = float(e2_value.get())*35.274
24
25     # Enters the converted weight to
26     # the text widget
27     t1.delete("1.0", END)
28     t1.insert(END, gram)
29
30     t2.delete("1.0", END)
31     t2.insert(END, pound)
32
33     t3.delete("1.0", END)
34     t3.insert(END, ounce)
35
36 # Create the Label widgets
37 e1 = Label(window, text = "Enter the weight in Kg")
```

```
38 e2_value = StringVar()
39 e2 = Entry(window, textvariable = e2_value)
40 e3 = Label(window, text = 'Gram')
41 e4 = Label(window, text = 'Pounds')
42 e5 = Label(window, text = 'Ounce')
43
44 # Create the Text Widgets
45 t1 = Text(window, height = 1, width = 20)
46 t2 = Text(window, height = 1, width = 20)
47 t3 = Text(window, height = 1, width = 20)
48
49 # Create the Button Widget
50 b1 = Button(window, text = "Convert", command = from_kg)
51
52 # grid method is used for placing
53 # the widgets at respective positions
54 # in table like structure
55 e1.grid(row = 0, column = 0)
56 e2.grid(row = 0, column = 1)
57 e3.grid(row = 1, column = 0)
58 e4.grid(row = 1, column = 1)
59 e5.grid(row = 1, column = 2)
60 t1.grid(row = 2, column = 0)
61 t2.grid(row = 2, column = 1)
62 t3.grid(row = 2, column = 2)
63 b1.grid(row = 0, column = 2)
64
65 # Start the GUI
66 window.mainloop()
67
```



Republic of the Philippines  
**CAMARINES SUR POLYTECHNIC COLLEGES**  
Nabua, Camarines Sur  
**COLLEGE of COMPUTER STUDIES**



## Modified Code

```
File Edit Selection View Go Run Terminal Help
C:\Users>justine>Downloads>Converter.py
1 #Weight and Height Converter by Group 6 - DEFENSIVE PROGRAMMING FINAL PROJECT
2 from tkinter import *
3 from tkinter import ttk
4 from collections import deque
5
6 class ConverterApp:
7     def __init__(self, master):
8         # OOP: Initialize the ConverterApp class
9         self.master = master
10        master.title("W&H-C-GR0UP6")
11        self.recent_results = deque(maxlen=10) # Keep track of the last 10 conversion results
12
13        # Set up the GUI
14        self.setup_gui()
15
16    def setup_gui(self):
17        # OOP: Window setup
18        self.master.geometry("645x850")
19        style = ttk.Style()
20        style.configure("TLabel", font=("Arial", 10))
21        style.configure("TButton", font=("Arial", 10), padding=(10, 5))
22        style.configure("TFrame", background="#f0f0f0")
23
24        # Create different sections of the GUI
25        self.create_header()
26        self.create_info_section()
27        # OOP: Method for creating the converter section
28        self.create_converter_section("Weight in kilogram(kg)", self.convert_weight, ["Gram(g)", "Pounds(lb)", "Ounce(oz)"])
29        self.create_converter_section("Height in feet (ft)", self.convert_height, ["Centimeters(cm)", "Meters(m)", "Inches(in)"])
30        self.create_footer()
31        self.create_recent_results_button()
32
33    def create_header(self):
34        # OOP: Method for creating header label
35        header_label = ttk.Label(
36            self.master,
37            text="Weight and Height Converter - GROUP 6",
```

```
38        style="TLabel",
39        font=("Helvetica", 16, "bold"),
40        background="#3498db",
41        foreground="white"
42    )
43    header_label.grid(row=0, column=0, columnspan=3, pady=10, sticky="ew")
44
45    def create_info_section(self):
46        # OOP: Method for creating info section
47        info_frame = ttk.Frame(padding="10", relief="groove", style="TFrame")
48        info_frame.grid(row=1, column=0, columnspan=2, padx=10, pady=10, sticky="nsew")
49
50        app_info_label = ttk.Label(
51            info_frame,
52            text="Weight and Height Converter\n\n"
53            "This app allows you to convert weights and heights between different units.\n\n"
54            "Follow the steps below to use the converter:\n\n"
55            "1. Enter the weight in kilograms and feet for height.\n\n"
56            "2. Click the 'Convert' button to see the conversion results.\n\n"
57            "3. Click the 'Clear' button to remove the current results.\n\n"
58            "4. Recent conversion results can also be seen by clicking \n the 'View Recent Conversion Results' button.\n\n"
59            "Accepted inputs:\n\n"
60            "- Positive numeric values\n\n"
61            "Not accepted inputs:\n\n"
62            "- Negative values\n\n"
63            "- Non-numeric values\n\n",
64            style="TLabel",
65            justify=LEFT
66        )
67        app_info_label.grid(row=0, column=0, padx=5, pady=5, sticky="nw")
68
69    def create_converter_section(self, label_text, convert_func, unit_labels):
70        # OOP: Method for creating converter section for weight or height
71        frame = ttk.Frame(self.master, padding="10", relief="groove", style="TFrame")
72        frame.grid(row=self.master.grid_size()[1], column=0, padx=10, pady=10, sticky="nsew")
73
74        label_prompt = ttk.Label(frame, text=f"Enter the {label_text.lower()}:", style="TLabel")
```



Republic of the Philippines  
**CAMARINES SUR POLYTECHNIC COLLEGES**  
Nabua, Camarines Sur



**COLLEGE of COMPUTER STUDIES**

```
File Edit Selection View Go Run Terminal Help
C:\Users> justine > Downloads > Converter.py > ConverterApp > create_converter_section

74 label_prompt = ttk.Label(frame, text=f"Enter the {label_text.lower()}:", style="TLabel")
75 entry_value = ttk.Entry(frame)
76 button_convert = ttk.Button(frame, text="Convert", command=lambda: convert_func(entry_value, result_widgets, error_label), style="TButton")
77 button_clear = ttk.Button(frame, text="Clear", command=lambda: self.clear_result_widgets(result_widgets), style="TButton")
78 error_label = ttk.Label(frame, text="", foreground="red", style="TLabel")
79
80 # Result Text widgets
81 result_widgets = [text(frame, height=1, width=20) for _ in unit_labels]
82
83 # Grid layout for Converter Section
84 label_prompt.grid(row=0, column=0, columnspan=2, pady=5, sticky="w")
85 entry_value.grid(row=0, column=2, padx=5, pady=5)
86 button_convert.grid(row=0, column=3, padx=5, pady=5)
87 button_clear.grid(row=1, column=3, padx=5, pady=5)
88 error_label.grid(row=3, column=0, columnspan=4, pady=5, sticky="w")
89
90 # Units labels and result text widgets
91 for i, unit_label in enumerate(unit_labels):
92     ttk.Label(frame, text=unit_label, style="TLabel").grid(row=2, column=i)
93     result_widgets[i].grid(row=1, column=i)
94
95 def create_footer(self):
96     # OOP: Method for creating footer label
97     footer_label = ttk.Label(self.master, text="© 2023 W&HC-GROUP6.", style="TLabel", background="#f0f0f0", font=("Arial", 8))
98     footer_label.grid(row=5, column=0, columnspan=3, pady=(10, 5), sticky="ew")
99
100 def create_recent_results_button(self):
101     # OOP: Method for creating button to view recent results
102     button_view_results = ttk.Button(self.master, text="View Recent Conversion Results", command=self.view_recent_results, style="TButton")
103     button_view_results.grid(row=4, column=0, columnspan=3, pady=10)
104
105 def convert_weight(self, entry_value, result_widgets, error_label):
106     # Input Validation: Check for non-negative numeric value
107     self.convert(entry_value, result_widgets, error_label, "Weight(kg)", [1000, 2.20462, 35.274])
108
109 def convert_height(self, entry_value, result_widgets, error_label):
110     # Input Validation: Check for non-negative numeric value
```

```
File Edit Selection View Go Run Terminal Help
C:\Users> justine > Downloads > Converter.py > ConverterApp > create_converter_section

111 self.convert(entry_value, result_widgets, error_label, "Height(ft)", [30.48, 0.3048, 12])
112
113 def convert(self, entry_value, result_widgets, error_label, label_text, conversion_factors):
114     # Input validation, error handling, and actual conversion logic
115     try:
116         value_str = entry_value.get().strip()
117         assert value_str, "Enter a number to convert."
118         value = float(value_str)
119         assert value >= 0, f"{label_text} cannot be negative."
120
121         results = [value * factor for factor in conversion_factors]
122
123         for widget, val in zip(result_widgets, results):
124             self.update_result_widgets(widget, val)
125
126         result_str = f"{label_text}: {value:.2f}\n"
127
128         # Define units for each type of conversion
129         unit_labels = ["", "Centimeters(cm)", "Meters(m)", "Inches(in)"] if "Height" in label_text else ["", "Gram(g)", "Pounds(lb)", "Ounces(oz)"]
130
131         # Create a list of formatted strings for each unit and result
132         formatted_results = [f"{unit:<15}: {res:.2f}" for res, unit in zip(results, unit_labels[1:])]
133
134         # Modify the format of the result_str concatenation
135         result_str += "\n".join(formatted_results)
136         result_str += "\n" + "-" * 30 # Line separator
137         self.recent_results.append(result_str)
138
139     except ValueError as error:
140         # Error Exception: Handle ValueError during conversion
141         self.show_error_message(error, error_label)
142     except AssertionError as error:
143         # Assert: Handle AssertionError for input validation
144         self.show_error_message(error, error_label)
145
146 def update_result_widgets(self, result_widget, value):
```



Republic of the Philippines  
**CAMARINES SUR POLYTECHNIC COLLEGES**  
Nabua, Camarines Sur



**COLLEGE of COMPUTER STUDIES**

```
147 def update_result_widgets(self, result_widget, value):
148     # Output Encoding: Update the result widgets with the calculated values
149     result_widget.delete("1.0", END)
150     result_widget.insert(END, f"{value:.2f}")
151
152 def clear_error_message(self, error_label):
153     # Clear the error message label
154     error_label.config(text="")
155
156 def clear_result_widgets(self, result_widgets):
157     # Clear the result widgets
158     for widget in result_widgets:
159         widget.delete("1.0", END)
160
161 def show_error_message(self, error, error_label):
162     # Error Handling: Display error message
163     error_label.config(text=f"Error: {error}")
164
165 def view_recent_results(self):
166     # Decorators: Display the recent conversion results in a new window
167     recent_results_window = Toplevel(self.master)
168     recent_results_window.title("Recent Conversion Results")
169
170     results_text = Text(recent_results_window, height=30, width=30)
171     results_text.pack(padx=10, pady=10)
172
173     for result in reversed(self.recent_results):
174         results_text.insert(END, result + "\n")
175
176 if __name__ == "__main__":
177     # OOP: Create the Tkinter root window and run the application
178     root = Tk()
179     app = ConverterApp(root)
180     root.mainloop()
```

## Original Code



## COLLEGE *of* COMPUTER STUDIES

### Modified Code

#### ADDITIONAL FEATURES: INSTRUCTIONS, HEIGHT CONVERSION AND RECENT RESULTS

W&HC-GROUP6

### Weight and Height Converter - GROUP 6

**Weight and Height Converter**

This app allows you to convert weights and heights between different units.

Follow the steps below to use the converter:

1. Enter the weight in kilograms and feet for height.
2. Click the 'Convert' button to see the conversion results.
3. Click the 'Clear' button to remove the current results.
4. Recent conversion results can also be seen by clicking the 'View Recent Conversion Results' button.

**Accepted inputs:**

- Positive numeric values

**Not accepted inputs:**

- Negative values
- Non-numeric values

Enter the weight in kilogram(kg):

Convert

12000.00	26.46	423.29
Gram(g)	Pounds(lb)	Ounce(oz)

Clear

Enter the height in feet (ft):

Convert

182.88	1.83	72.00
Centimeters(cm)	Meters(m)	Inches(in)

Clear

View Recent Conversion Results

© 2023 W&HC-GROUP6.

Recent Conversion Results

Height (ft) :	6.00
Centimeters (cm) :	182.88
Meters (m) :	1.83
Inches (in) :	72.00
=====	
Weight (kg) :	12.00
Gram (g) :	12000.00
Pounds (lb) :	26.46
Ounce (oz) :	423.29
=====	



## COLLEGE *of* COMPUTER STUDIES

### INPUT VALIDATION and ERROR HANDLING

-TRYING TO INPUT A NEGATIVE VALUE

Enter the weight in kilogram(kg):

-12

Convert

Clear

Gram(g)

Pounds(lb)

Ounce(oz)

Error: Weight(kg) cannot be negative.

Enter the height in feet (ft):

-6

Convert

Clear

Centimeters(cm)

Meters(m)

Inches(in)

Error: Height(ft) cannot be negative.

View Recent Conversion Results

© 2023 W&HC-GROUP6.

-TRYING TO INPUT NON NUMERIC VALUE

Enter the weight in kilogram(kg):

@@

Convert

Clear

Gram(g)

Pounds(lb)

Ounce(oz)

Error: could not convert string to float: '@@'

Enter the height in feet (ft):

qwerty

Convert

Clear

Centimeters(cm)

Meters(m)

Inches(in)

Error: could not convert string to float: 'qwerty'

View Recent Conversion Results

© 2023 W&HC-GROUP6.





## **COLLEGE of COMPUTER STUDIES**

-TRYING TO INPUT EMPTY VALUE

The screenshot displays a graphical user interface for a conversion application. It is divided into two main sections: weight conversion and height conversion. The weight section has a label 'Enter the weight in kilogram(kg):' followed by an empty text input field, a 'Convert' button, and three radio buttons for 'Gram(g)', 'Pounds(lb)', and 'Ounce(oz)'. Below these is a red error message: 'Error: Enter a number to convert.' The height section has a label 'Enter the height in feet (ft):' followed by an empty text input field, a 'Convert' button, and three radio buttons for 'Centimeters(cm)', 'Meters(m)', and 'Inches(in)'. Below these is another red error message: 'Error: Enter a number to convert.' At the bottom of the interface is a 'View Recent Conversion Results' button and a copyright notice '© 2023 W&HC-GROUP6.'

#### **IV. Results**

As a result of this modification, the weight converter program has undergone significant improvements. The code has been restructured using object-oriented programming, resulting in enhanced organization and readability. The graphical user interface (GUI) now features a design, including instructions, footer, clear buttons with and a convenient button for viewing recent conversion results. The application supports both weight and height conversion, with a more dynamic and modular approach. Clear error messages are provided to users for better input validation. Overall, these modifications have transformed the program into a user-friendly tool for converting weights and heights, with improved design and functionality.

#### **V. Reference**

**Original source code:**

<https://www.geeksforgeeks.org/python-weight-conversion-gui-using-tkinter>