



# **PES UNIVERSITY**

**(Established under Karnataka Act No. 16 of 2013)**

**100 Feet Ring Road, BSK III Stage, Bengaluru-560 085**

**Department of Computer Science & Engineering**

## **Title: Unit Converter**

**Team member details: PES1202100489 - Vandana S**

**PES1202101272 - Trisha Songra**

**PES1202101372 - Uttam S Gowda**

## **Abstract**

This project is a representation of a simple unit converter made using Python programming language. It makes use of functions, GUI, dictionaries, lists and basic python's constructs namely the if-else construct. The purpose of this project is to allow the user to convert from one metric unit of the SI system to another with ease using graphical user interface. The output is produced on a simple GUI window where it allows the user to select the input unit, the output unit and enter the value which they would like to convert from.

## Table Of Contents

<b>Page No.</b>	<b>Content</b>
<b>1</b>	<b>Abstract</b>
<b>2</b>	<b>Table Of Contents</b>
<b>3</b>	<b>Introduction</b>
<b>4</b>	<b>Design/Implementation</b>
<b>5</b>	<b>Testing</b>
<b>6</b>	<b>Result And Analysis</b>
<b>7</b>	<b>Conclusion And Future Enhancements</b>
<b>8</b>	<b>References</b>

## **Introduction**

The problem our team has solved in this project is that of a Unit Converter. This result obtained allows the user to cruise through the various units in our metric system and also convert them from one to the other. The output of the code is represented using graphical user interface on a window consisting of multiple input required from the user. The user must select the unit they would like to enter the input, and select the unit they want the output in. Once they input the desired value, click on the button “CONVERT”, the corresponding output is displayed. There are a wide type of units to choose from. This simplifies the conversion of one metric unit to the other using GUI. The main purpose of GUI is to un-complicate the input given by the user which is what it has done in the corresponding project. There have been many components of Python used in the project such as lists, dictionaries, basic constructs i.e. if-else construct.

## Design/Implementation :

```
from tkinter import *
```

```
# Conversion factors
```

```
unit_dict = {  
    "cm": 0.01,  
    "m": 1.0,  
    "km": 1000.0,  
    "feet": 0.3048,  
    "miles": 1609.344,  
    "inches": 0.0254,  
    "grams": 1.0,  
    "kg": 1000.0,  
    "quintals": 100000.0,  
    "tonnes": 1000000.0,  
    "pounds": 453.592,  
    "sq. m": 1.0,  
    "sq. km": 1000000.0,  
    "are": 100.0,  
    "hectare": 10000.0,  
    "acre": 4046.856,  
    "sq. mile": 2590000.0,  
    "sq. foot": 0.0929,  
    "cu. cm": 0.001,  
    "litre": 1.0,  
    "ml": 0.001,  
    "gallon": 3.785  
}
```

```
lengths = ["cm", "m", "km", "feet", "miles", "inches"]
```

```
weights = ["kg", "grams", "quintals", "tonnes", "pounds"]
```

```
temps = ["Celsius", "Fahrenheit"]
```

```
areas = ["sq. m", "sq. km", "are", "hectare", "acre", "sq. mile", "sq. foot"]
```

```
volumes = ["cu. cm", "litre", "ml", "gallon"]
```

```
# Options for drop-down menu
```

```
OPTIONS = ["select units",  
    "cm",  
    "m",
```

```
"km",
"feet",
"miles",
"inches",
"kg",
"grams",
"quintals",
"tonnes",
"pounds",
"Celsius",
"Fahrenheit",
"sq. m",
"sq. km",
"are",
"hectare",
"acre",
"sq. mile",
"sq. foot",
"cu. cm",
"litre",
"ml",
"gallon"]

# Main window
root = Tk()

root.geometry("600x430")
root.title("UNIVERT")
root['bg'] = 'black'

def ok():
    inp = float(inputentry.get())
    inp_unit = inputopt.get()
    out_unit = outputopt.get()

    cons = [inp_unit in lengths and out_unit in lengths,
            inp_unit in weights and out_unit in weights,
            inp_unit in temps and out_unit in temps,
            inp_unit in areas and out_unit in areas,
            inp_unit in volumes and out_unit in volumes]

    if any(cons): # If both the units are of same type, do the conversion
```

```
if inp_unit == "Celsius" and out_unit == "Fahrenheit":
    outputentry.delete(0, END)
    outputentry.insert(0, (inp * 1.8) + 32)
elif inp_unit == "Fahrenheit" and out_unit == "Celsius":
    outputentry.delete(0, END)
    outputentry.insert(0, (inp - 32) * (5/9))
else:
    outputentry.delete(0, END)
    outputentry.insert(0, round(inp * unit_dict[inp_unit]/unit_dict[out_unit], 5))

else: # Display error if units are of different types
    outputentry.delete(0, END)
    outputentry.insert(0, "ERROR")

inputopt = StringVar()
inputopt.set(OPTIONS[0])

outputopt = StringVar()
outputopt.set(OPTIONS[0])

# Widgets
inputlabel = Label(root, text="FROM")
inputlabel.grid(row=0, column=0, pady=20)
inputlabel.config(font="Pushster")
inputlabel['bg'] = 'black'

inputentry = Entry(root, justify="center", font="bold",
width=40)
inputentry.grid(row=1, column=0, padx=35, ipady=5)
inputentry.config(font="Pushster")
inputentry['bg'] = 'black'

inputmenu = OptionMenu(root, inputopt, *OPTIONS)
inputmenu.grid(row=1, column=1)
inputmenu.config(font="Pushster")
inputmenu['bg'] = 'black'

outputlabel = Label(root, text="TO")
outputlabel.grid(row=4, column=0, pady=20)
```

```
outputlabel['bg'] = 'black'
```

```
outputentry = Entry(root, justify="center", font="bold",  
width=40)
```

```
outputentry.grid(row=5, column=0, padx=35, pady=5)
```

```
outputentry.config(font="Pushster")
```

```
outputentry['bg'] = 'black'
```

```
outputmenu = OptionMenu(root, outputopt, *OPTIONS)
```

```
outputmenu.grid(row=5, column=1)
```

```
outputmenu.config(font="Pushster")
```

```
outputmenu['bg'] = 'black'
```

```
okbtn = Button(root, text="CONVERT", command=ok, padx=80, pady=2)
```

```
okbtn.grid(row=3, column=0, columnspan=2, pady=50)
```

```
okbtn.config(font="Pushster")
```

```
okbtn['bg'] = 'black'
```

```
root.mainloop()
```



## Testing

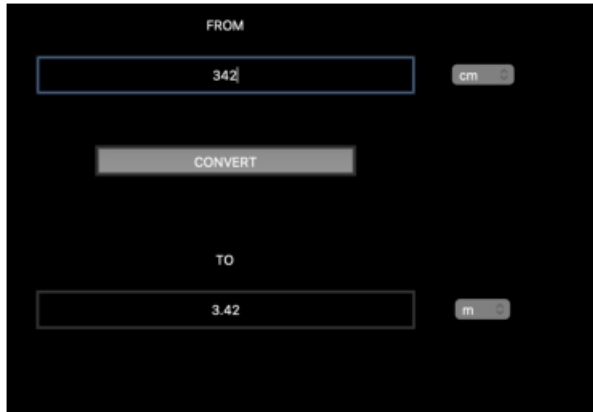
There were 7 test cases -

Test Cases	Test Results
1 - cm to m	Input conversion successful.
2 - kg to pounds	Input conversion successful.
3 - *C to *F	Input conversion successful.
4 - sq. m to acre	Input conversion successful.
5 - l to gallon	Input conversion successful.
6 - cm to cm	Output same as Input.
7 - cm to kg	“ERROR” displayed as conversion is not possible.



\*Error obtained was rectified as per lab in-charge indication.

## Results And Analysis



FROM

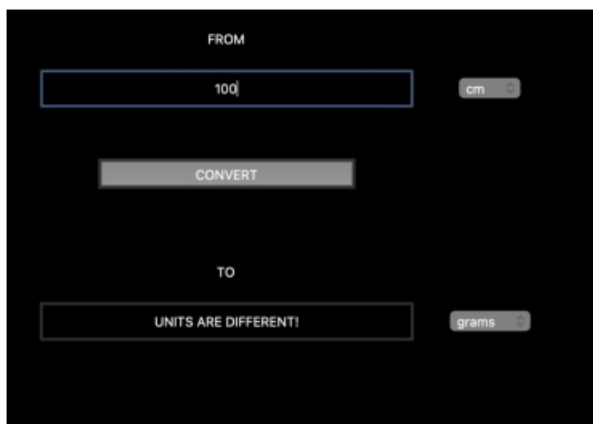
342 cm

CONVERT

TO

3.42 m

1. When Input Unit and Output Unit are in the same category -



FROM

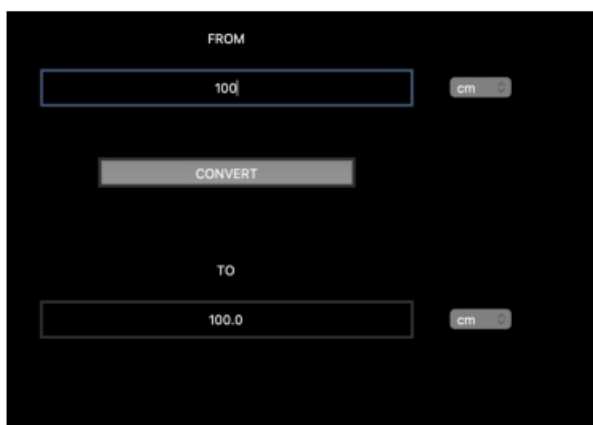
100 cm

CONVERT

TO

UNITS ARE DIFFERENT! grams

2. When Input Unit and Output Unit are in not the same category -



FROM

100 cm

CONVERT

TO

100.0 cm

3. When Input Unit and Output Unit are the same -

## Conclusions & Future Enhancements

To conclude, Project Unit Converter works like a component which can convert units from one metric system to the other. It overcomes the many limitations incorporated in the attendance.

- Easy usability feature.
- Generate precise output.

This project has a vast scope in the future and can be implemented to the web applications due to its flexibility of upgradation to include any and all necessary classes of units to expand its scope. With the programmed software of Unit Converter, the user is able to easily simplify the needs of conversion in a simple, accurate and error free manner. The following are the future scope for the project.

- Implementation in Word Processing Software.
- Implementation in Data Management System.
- Expansion to various other fields, namely Photo Processing etc.