**Laney College**

**Computer Information Systems (CIS) Department**

**Programming Assignment Cover Sheet**

**Class: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Email: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Project Title: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Problem Description/Purpose**

This purpose of the program is to convert a length unit to another length unit. The user can able to input a digit and select the options for the converting. For example, if the user enter a digit and select the option converting the centimeter to meter, the program will do a calculation and print out the meter output.

**Design**

About my program design, there have two files. The first file is a header that includes the calculation. Inside the header, it has four classes such as centimeter, kilometer, meter, and millimeter. And inside each of the class, it has different functions for the converting. For example, centimeter will have some functions like “To meter”, “To kilometer”, or “To millimeter.” And the other classes has the same structure.

**User Interface**

My second file is a user interface, which build by a standard GUI package call Tkinter. The user interface includes different operators. One of the operators is the radio button that the user will select the length unit and the length unit that the user want to convert. In addition, these radio buttons would only allow the user select one on each sides. One of the operators is the entry that the user can able input the digit. Another operators are the buttons such as the convert button that when the user press it, it can do the conversion. And a exit button that when the user press it, it will exit the program.

**Header File**

class Centimeters:

def \_\_init\_\_(self, unit):

self.Unit = unit

def ToMeters(self):

return int(self.Unit) / 100

def ToMillimeters(self):

return int(self.Unit) \* 10

def ToKilometers(self):

return int(self.Unit) / 100000

def ToYards(self):

return self.Unit \* 0.0109361

def ToFeets(self):

return self.Unit \* 0.0328084

def ToInches(self):

return self.Unit \* 0.393701

def print(self):

print("Unit", self.Unit)

class Millimeters:

def \_\_init\_\_(self, unit):

self.Unit = unit

def ToMeters(self):

return int(self.Unit) / 1000

def ToCentimeters(self):

return int(self.Unit) / 10

def ToKilometers(self):

return int(self.Unit) / 1000000

def ToYards(self):

return self.Unit \* 0.00109361

def ToFeets(self):

return self.Unit \* 0.00328084

def ToInches(self):

return self.Unit \* 0.0393701

def print(self):

print("Unit", self.Unit)

class Meters:

def \_\_init\_\_(self, unit):

self.Unit = unit

def ToCentimeters(self):

return int(self.Unit) \* 100

def ToMillimeters(self):

return int(self.Unit) \* 1000

def ToKilometers(self):

return int(self.Unit) / 1000

def ToYards(self):

return self.Unit \* 1.09361

def ToFeets(self):

return self.Unit \* 3.28084

def ToInches(self):

return self.Unit \* 39.3701

def print(self):

print("Unit", self.Unit)

class Kilometers:

def \_\_init\_\_(self, unit):

self.Unit = unit

def ToCentimeters(self):

return int(self.Unit) \* 100000

def ToMillimeters(self):

return int(self.Unit) \* 1000000

def ToMeters(self):

return int(self.Unit) \* 1000

def ToYards(self):

return self.Unit \* 1093.61

def ToFeets(self):

return self.Unit \* 3280.84

def ToInches(self):

return self.Unit \* 39370.1

def print(self):

print("Unit", self.Unit)

class Yards:

def \_\_init\_\_(self, unit):

self.Unit = unit

def ToCentimeters(self):

return self.Unit \* 91.44

def ToMillimeters(self):

return self.Unit \* 914.4

def ToMeters(self):

return self.Unit \* 0.9144

def ToKilometers(self):

return self.Unit \* 0.0009144

def ToFeets(self):

return self.Unit \* 3

def ToInches(self):

return self.Unit \* 36

def print(self):

print("Unit", self.Unit)

class Feets:

def \_\_init\_\_(self, unit):

self.Unit = unit

def ToCentimeters(self):

return self.Unit \* 30.48

def ToMillimeters(self):

return self.Unit \* 304.8

def ToMeters(self):

return self.Unit \* 0.3048

def ToYards(self):

return self.Unit \* 0.333333

def ToKilometers(self):

return self.Unit \* 0.0003048

def ToInches(self):

return self.Unit \* 12

def print(self):

print("Unit", self.Unit)

class Inches:

def \_\_init\_\_(self, unit):

self.Unit = unit

def ToCentimeters(self):

return self.Unit \* 2.54

def ToMillimeters(self):

return self.Unit \* 25.4

def ToMeters(self):

return self.Unit \* 0.0254

def ToYards(self):

return self.Unit \* 0.0277778

def ToFeets(self):

return self.Unit \* 0.0833333

def ToKilometers(self):

return self.Unit \* 0.0000254

def print(self):

print("Unit", self.Unit)

**Interface File**

from tkinter import \*

import tkinter.messagebox

from ProjectHeader import \*

class Gui(object):

def \_\_init\_\_(self, parent):

self.gui = parent

self.gui.geometry("350x200")

self.gui.title("Length Converter")

self.checked1 = IntVar()

self.checked2 = IntVar()

self.c1 = Radiobutton(self.gui, text='(b1)Centimeters', variable=self.checked1, value = 1)

self.c2 = Radiobutton(self.gui, text='(b2)Meter', variable=self.checked1, value = 2)

self.c3 = Radiobutton(self.gui, text='(b3)Millimeters', variable=self.checked1, value = 3)

self.c4 = Radiobutton(self.gui, text='(b4)Kilometers', variable=self.checked1, value = 4)

self.c5 = Radiobutton(self.gui, text='(b5)Centimeters', variable=self.checked2, value = 5)

self.c6 = Radiobutton(self.gui, text='(b6)Meter', variable=self.checked2, value = 6)

self.c7 = Radiobutton(self.gui, text='(b7)Millimeters', variable=self.checked2, value = 7)

self.c8 = Radiobutton(self.gui, text='(b8)Kilometers', variable=self.checked2, value = 8)

self.b1 = Button(self.gui, text="Convert", command=self.callback)

self.b2 = Button(self.gui, text="Exit", command=self.exit)

self.l1 = Label(self.gui, text="Value")

self.l2 = Label(self.gui, text="Convert ->")

self.e1 = Entry(self.gui, bd = 5)

self.e1.insert(0, "0")

self.c1.pack(side = BOTTOM)

self.c2.pack(side = BOTTOM)

self.c3.pack(side = BOTTOM)

self.c4.pack(side = BOTTOM)

self.c5.pack(side = BOTTOM)

self.c6.pack(side = BOTTOM)

self.c7.pack(side = BOTTOM)

self.c8.pack(side = BOTTOM)

self.l1.pack(side = TOP)

self.l2.pack(side = TOP)

self.e1.pack(side = TOP)

self.b1.pack(side = TOP)

self.b2.pack(side = TOP)

self.c1.place(x = 3, y = 90)

self.c2.place(x = 3, y = 110)

self.c3.place(x = 3, y = 130)

self.c4.place(x = 3, y = 150)

self.c5.place(x = 235, y = 90)

self.c6.place(x = 235, y = 110)

self.c7.place(x = 235, y = 130)

self.c8.place(x = 235, y = 150)

self.l2.place(x = 140, y = 110)

self.b1.place(x = 107, y = 50)

self.b2.place(x = 210, y = 50)

def exit(self):

self.gui.destroy()

def callback(self):

if self.e1.get() == "0":

messagebox.showinfo("Error", "Please enter a value")

elif self.checked1.get() == 1: #Centimeters

C = Centimeters(self.e1.get())

if self.checked2.get() == 5:

messagebox.showinfo("Value", str(self.e1.get()) + ' cm')

if self.checked2.get() == 6:

value = C.ToMeters()

messagebox.showinfo("Value", str(value) + ' m')

if self.checked2.get() == 7:

value = C.ToMillimeters()

messagebox.showinfo("Value", str(value) + ' mm')

if self.checked2.get() == 8:

value = C.ToKilometers()

messagebox.showinfo("Value", str(value) + ' km')

elif self.checked1.get() == 2: #Meters

M = Meters(self.e1.get())

if self.checked2.get() == 5:

value = M.ToCentimeters()

messagebox.showinfo("Value", str(value) + ' cm')

if self.checked2.get() == 6:

messagebox.showinfo("Value", str(self.e1.get()) + ' m')

if self.checked2.get() == 7:

value = M.ToMillimeters()

messagebox.showinfo("Value", str(value) + ' mm')

if self.checked2.get() == 8:

value = M.ToKilometers()

messagebox.showinfo("Value", str(value) + ' km')

elif self.checked1.get() == 3: #Millimeters

Mi = Millimeters(self.e1.get())

if self.checked2.get() == 5:

value = Mi.ToCentimeters()

messagebox.showinfo("Value", str(value) + ' cm')

if self.checked2.get() == 6:

value = Mi.ToMeters()

messagebox.showinfo("Value", str(value) + ' m')

if self.checked2.get() == 7:

messagebox.showinfo("Value", str(self.e1.get()) + ' mm')

if self.checked2.get() == 8:

value = Mi.ToKilometers()

messagebox.showinfo("Value", str(value) + ' km')

elif self.checked1.get() == 4: #Kilometers

K = Kilometers(self.e1.get())

if self.checked2.get() == 5:

value = K.ToCentimeters()

messagebox.showinfo("Value", str(value) + ' cm')

if self.checked2.get() == 6:

value = K.ToMeters()

messagebox.showinfo("Value", str(value) + ' m')

if self.checked2.get() == 7:

value = K.ToMillimeters()

messagebox.showinfo("Value", str(value) + ' mm')

if self.checked2.get() == 8:

messagebox.showinfo("Value", str(self.e1.get()) + ' km')

else:

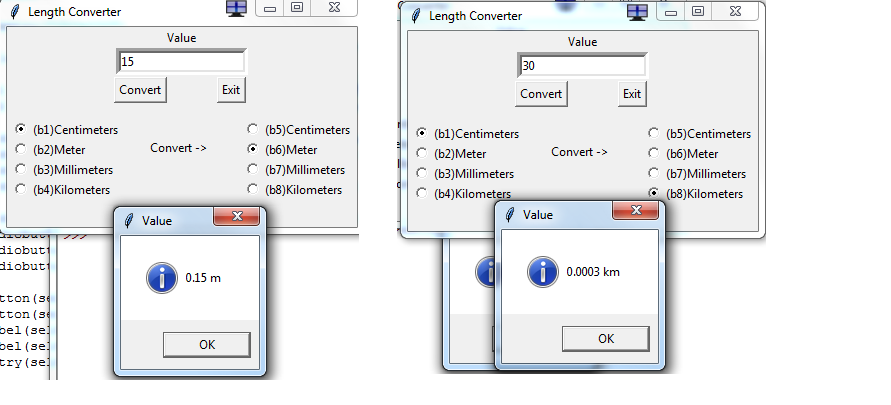
messagebox.showinfo("Error", "Please Select the options")

root = Tk()

my\_window = Gui(root)

root.mainloop()

**OUTPUT**

****