# CA 207 – Python lab


## Topic : Convert Number System.

( Prepared by : ***MANDEEP KUMAR***)
[ Guided by : ***Prof. Anirban Choudhary*** ]

- **NAME : MANDEEP KUMAR**
- **Scholor No. : 212120139**
- **Subject code: 107**
- **Stream : MCA**
- **Semester : 2nd**

# 2. Contents

# Convert Number System

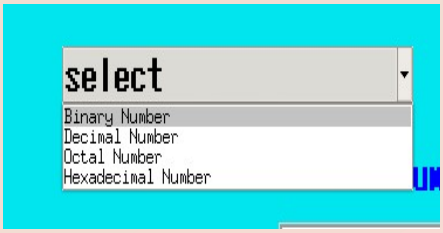
## Tool used

### 1. Combo box

A combobox is a combination of an Entry widget and a Listbox widget. A combobox widget allows you to select one value in a set of values. In addition, it allows you to enter a custom value.

**ex-**

```
var2=StringVar()
v2=["Binary Number","Decimal Number","Octal Number","Hexadecimal Number"]
combo2= ttk.Combobox(root,width=20,value=v2,font="Verdana 20
bold",textvariable=var2)
combo2['state']='readonly'
combo2.set("select")
combo2.place(x=550,y=100)
```




### 2. Level

Tkinter Label is a widget that is used to implement display boxes where you can place text or images. The text displayed by this widget can be changed by the developer at any time you want. It is also used to

perform tasks such as to underline the part of the text and span the text across multiple lines

```
no=Label(root,text="Enter Number:-->",bg="turquoise2",fg="blue",font="Verdana 20 bold")
no.place(x=350,y=160)
```


## 3. Button

The Button widget is used to add buttons in a Python application. These buttons can display text or images that convey the purpose of the buttons. You can attach a function or a method to a button which is called automatically when you click the button.

### Syntax

Here is the simple syntax to create this widget −

```
w = Button ( master, option=value, ... )
```

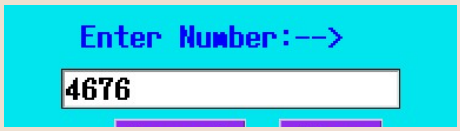
```
convert=Button(root,text="Convert",command=convert,font="Verdana 14 bold",bg="purple2")
convert.place(x=380,y=250)

clear=Button(root,text="Clear",command=clear ,font="Verdana 14 bold",bg="purple2")
clear.place(x=520,y=250)
```


## 4. Text Box

Tkinter Text box widget is used to insert multi-line text. This widget can be used for messaging, displaying information, and many other tasks. The important task is to get the inserted text for further processing. For this, we have to use the **get()** method for the textbox widget.

*Syntax: get(start, [end])*

```
num=StringVar()
ent=Entry(root,bg="white",bd=2,width=20,font="Verdana 20 bold",textvariable=num)
ent.place(x=335,y=205)
```




## 5. Messege box

**messagebox.showerror("error"," please.. select Anything ,Then try ...")**

***def* messege():**
**messagebox.showerror("error"," Base error,ENTER number must be less than base ")**

## Code

```
from tkinter import*
from tkinter import ttk
from tkinter import messagebox


def messege():
messagebox.showerror("error"," Base error,ENTER number must be less than
base ")

def dec_to_any_no(num, base,base1):
base_num = ""
for i in num: #validation
if ord(i)-ord('0')>=0 and ord(i)-ord('0')<base1:
continue
else :
messege() #validation
return base_num
n=int(num)
while n>0:
dig = int(n%base)
if dig<10:
base_num += str(dig)
else:
base_num += chr(ord('A')+dig-10)
n //= base
base_num = base_num[::-1]
return base_num



def any_to_dec_no(num,base):
for i in num: #validation
if ord(i)-ord('0')>=0 and ord(i)-ord('0')<base:
continue
else :
```

```
messege() #validation
return " "
temp=int(num,base)
return temp
def OctToHex(num):
for i in num: #validation
if ord(i)-ord('0')>=0 and ord(i)-ord('0')<8:
continue
else :
messege() #validation
return " "
return hex(int(num,8))
def oct_to_bin(num):
for i in num: #validation
if ord(i)-ord('0')>=0 and ord(i)-ord('0')<8:
continue
else :
messege() #validation
return " "
return bin(int(num,8))
def hextobin(num) :
return bin(int(num,16))
def bin_to_oct(num):
for i in num: #validation
if ord(i)-ord('0')>=0 and ord(i)-ord('0')<2:
continue
else :
messege() #validation
return " "
return oct(int(num,2))
def bin_to_hex(num):
for i in num: #validation
if ord(i)-ord('0')>=0 and ord(i)-ord('0')<2:
continue
else :
messege() #validation
return " "
return hex(int(num,2))
def hex_to_oct(num):
```

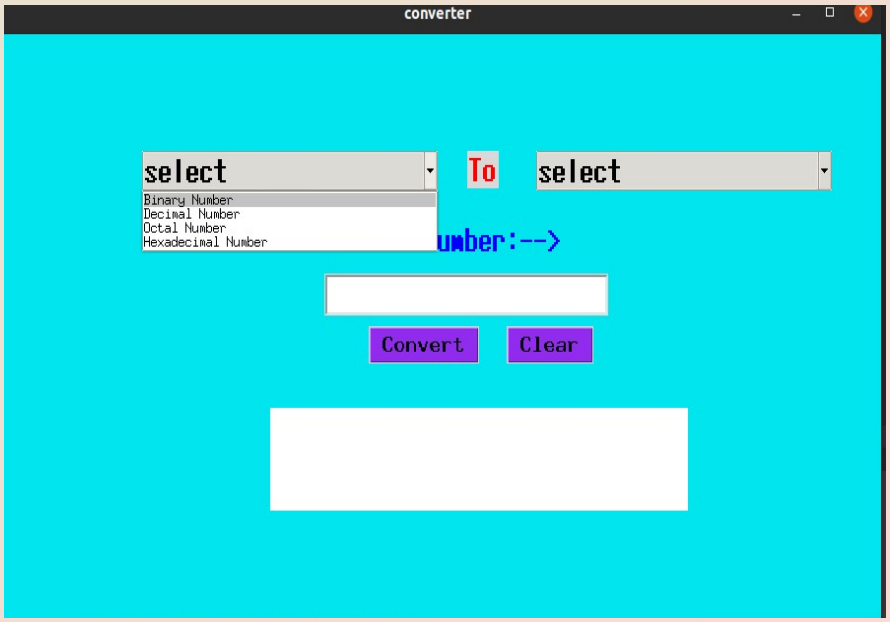
```
return oct(int(num,16))



def convert():
value1=var1.get()
value2=var2.get()
enter_val=num.get()
if value1=="select" and value2=="select" :
messagebox.showerror("error"," please.. select Anything ,Then try ...")
elif value1==value2 :
messagebox.showerror("error","select worng!, please do not select same field ")
elif enter_val=="" :
messagebox.showerror("error","please enter any number ,Then try .. ")
elif value1=="Decimal Number" and value2=="Binary Number":
val=dec_to_any_no(enter_val,2,10)
display.config(text=val)
elif value1=="Decimal Number" and value2=="Octal Number":
val=dec_to_any_no(enter_val,8,10)
display.config(text=val)
elif value1=="Decimal Number" and value2=="Hexadecimal Number":
val=dec_to_any_no(enter_val,16,10)
display.config(text=val)
elif value1=="Binary Number" and value2=="Decimal Number":
val=any_to_dec_no(enter_val,2)
display.config(text=val)

elif value1=="Binary Number" and value2=="Octal Number":
val=bin_to_oct(enter_val)
value=val[2:].upper()
display.config(text=value)

elif value1=="Binary Number" and value2=="Hexadecimal Number":
val=bin_to_hex(enter_val)
value=val[2:].upper()
display.config(text=value)

elif value1=="Octal Number" and value2=="Binary Number":
val=oct_to_bin(enter_val)
```

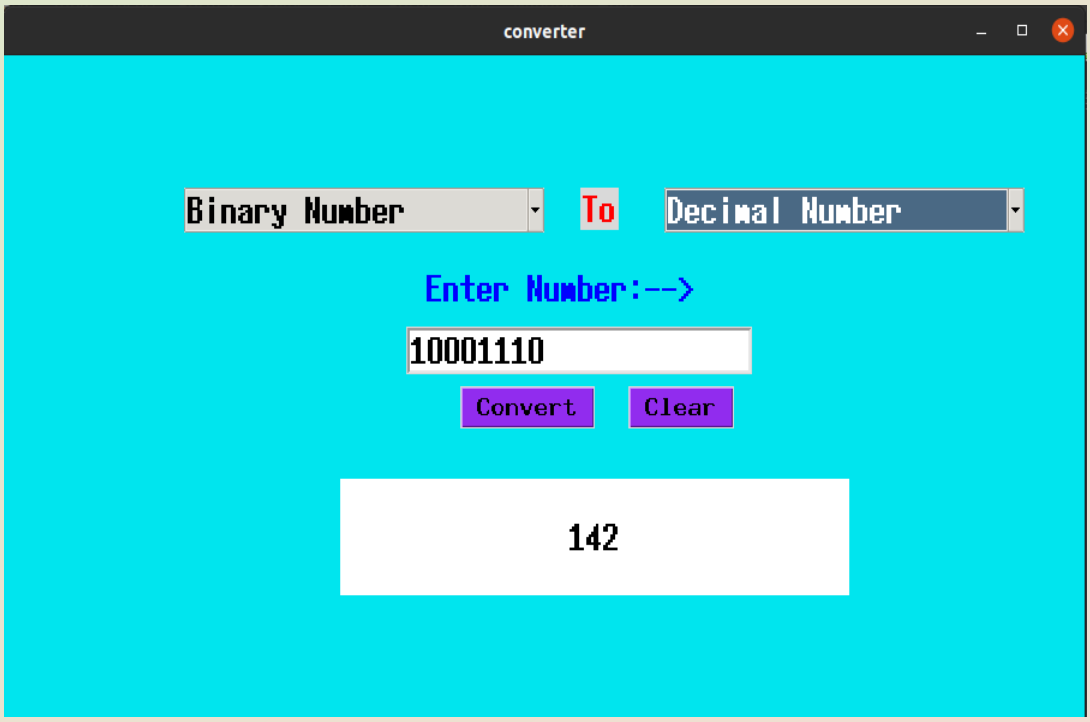
```
value=val[2:].upper()
display.config(text=value)

elif value1=="Octal Number" and value2=="Decimal Number":
val=any_to_dec_no(enter_val,8)
display.config(text=val)

elif value1=="Octal Number" and value2=="Hexadecimal Number":
val=OctToHex(enter_val)
value=val[2:].upper()
display.config(text=value)
elif value1=="Hexadecimal Number" and value2=="Binary Number":
val=hextobin(enter_val)
value=val[2:].upper()
display.config(text=value)
elif value1=="Hexadecimal Number" and value2=="Decimal Number":
val=any_to_dec_no(enter_val,16)
display.config(text=val)
elif value1=="Hexadecimal Number" and value2=="Octal Number":
val=hex_to_oct(enter_val)
value=val[2:].upper()
display.config(text=value)
else :
messagebox.showerror("error"," sorry! Wrong enter")

def clear() :
display.config(text="")
ent.delete(0, END)
combo1.set("select")
combo2.set("select")
root=Tk()
root.minsize(900, 500)
root.configure(bg='turquoise2')
root.title("converter")

style= ttk.Style()
style.theme_use('clam') # why combobox style works only when theme is
used???
style.configure("test1.TCombobox", fieldbackground= "white",)
```

```
var1=StringVar()
v1=["Binary Number","Decimal Number","Octal Number","Hexadecimal
Number"]
combo1= ttk.Combobox(root,width=20,value=v1,font="Verdana 20
bold" ,textvariable=var1)
combo1['state']='readonly'
combo1.set("select")
combo1.place(x=150,y=100)
combo1['style'] = "test1.TCombobox"

to=Label(root,text="To",fg="red",font="Verdana 20 bold")
to.place(x=480,y=100)

var2=StringVar()
v2=["Binary Number","Decimal Number","Octal Number","Hexadecimal
Number"]
combo2= ttk.Combobox(root,width=20,value=v2,font="Verdana 20
bold",textvariable=var2)
combo2['style'] = "test1.TCombobox"
combo2['state']='readonly'
combo2.set("select")
combo2.place(x=550,y=100)

no=Label(root,text="Enter Number:-->",bg="turquoise2",fg="blue",font="Verdana
20 bold")
no.place(x=350,y=160)

num=StringVar()
ent=Entry(root,bg="white",bd=2,width=20,font="Verdana 20
bold",textvariable=num)
ent.place(x=335,y=205)

convert=Button(root,text="Convert",command=convert,font="Verdana 14
bold",bg="purple2")
convert.place(x=380,y=250)
```

```
clear=Button(root,text="Clear",command=clear ,font="Verdana 14
bold",bg="purple2")
clear.place(x=520,y=250)

display=Label(root,text="
",width=30,height=3,fg="black",bg="white",font="Verdana 20 bold")
display.place(x=280,y=320)


root.mainloop()
```

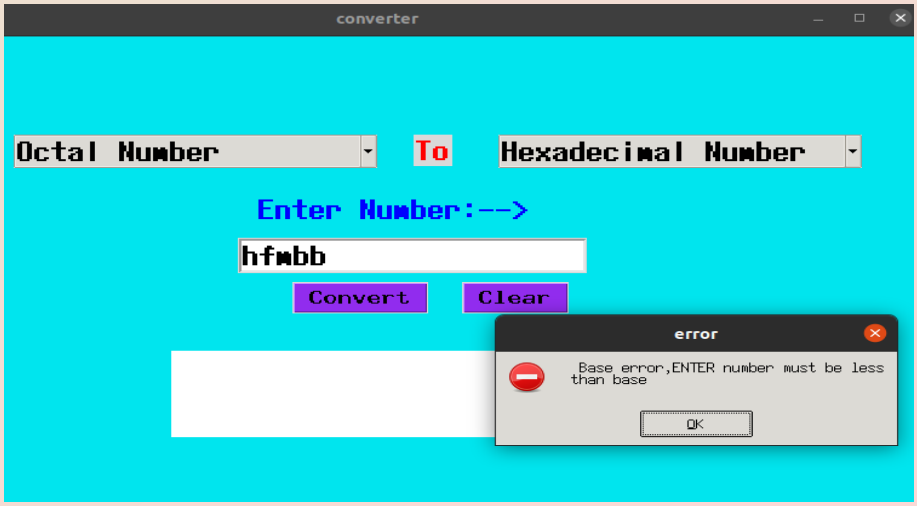
## OUTPUT



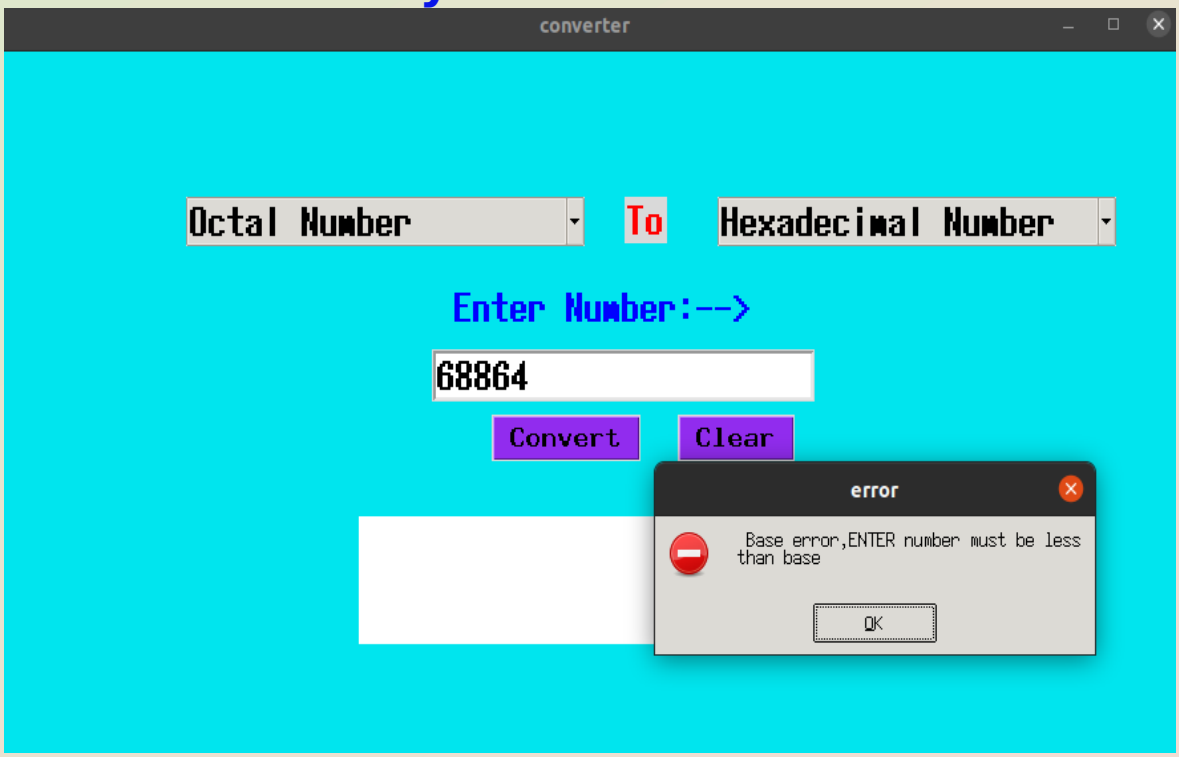



# VALIDATION AND ERROR

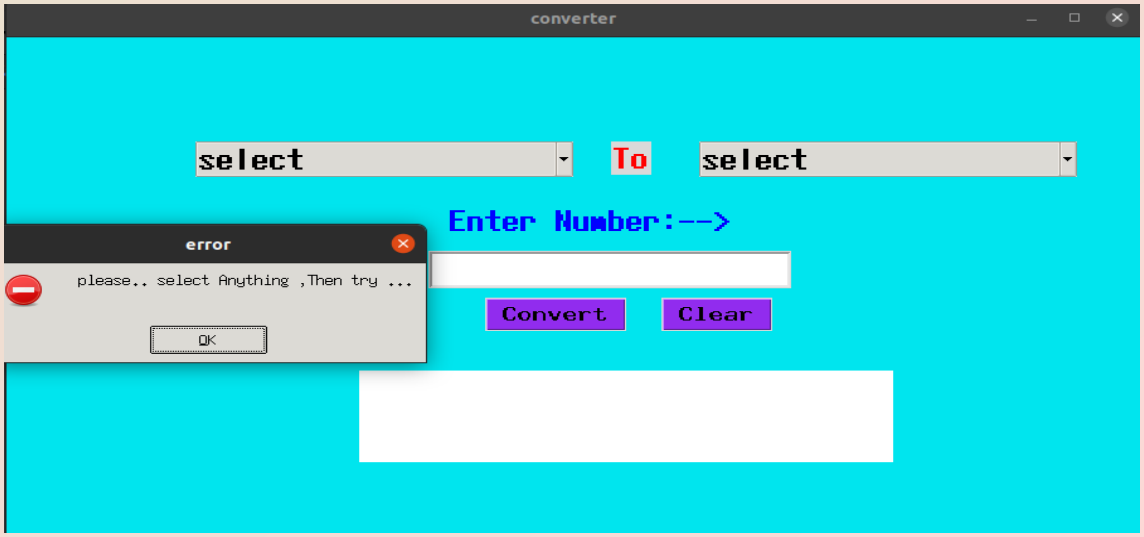
## a) Do not enter string in a text box



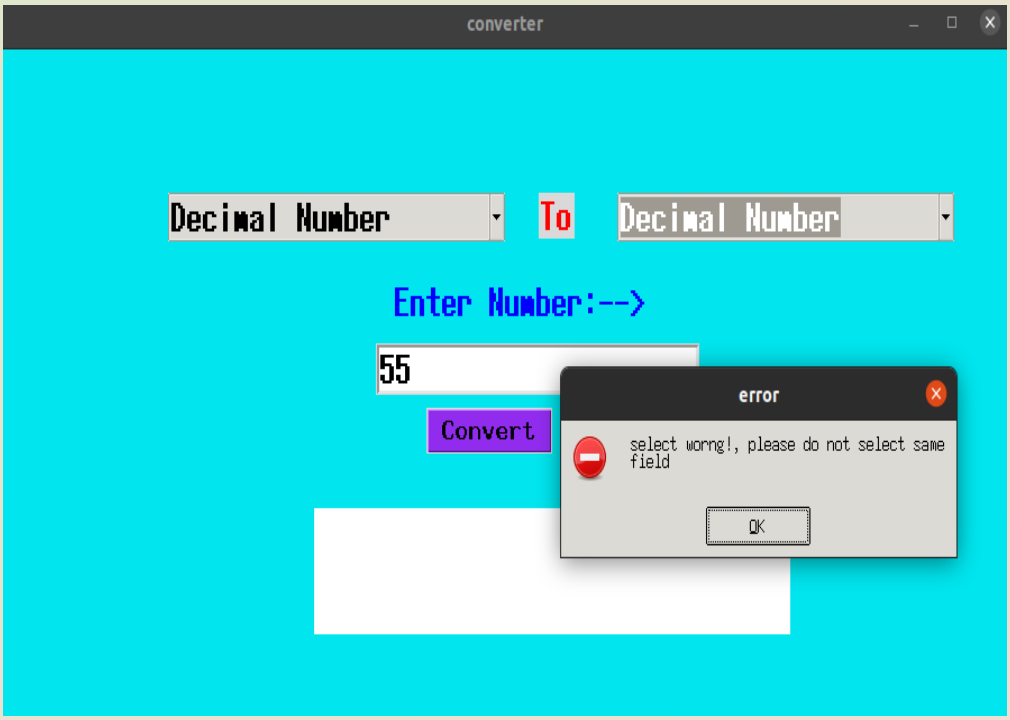
## b) Do not enter enter number more than base of number system




## c) Enter all fill to required.



d)


converter
Decimal Number
To
Decimal Number
Enter Number:-->
55
Convert
error
select worng!, please do not select same field
OK