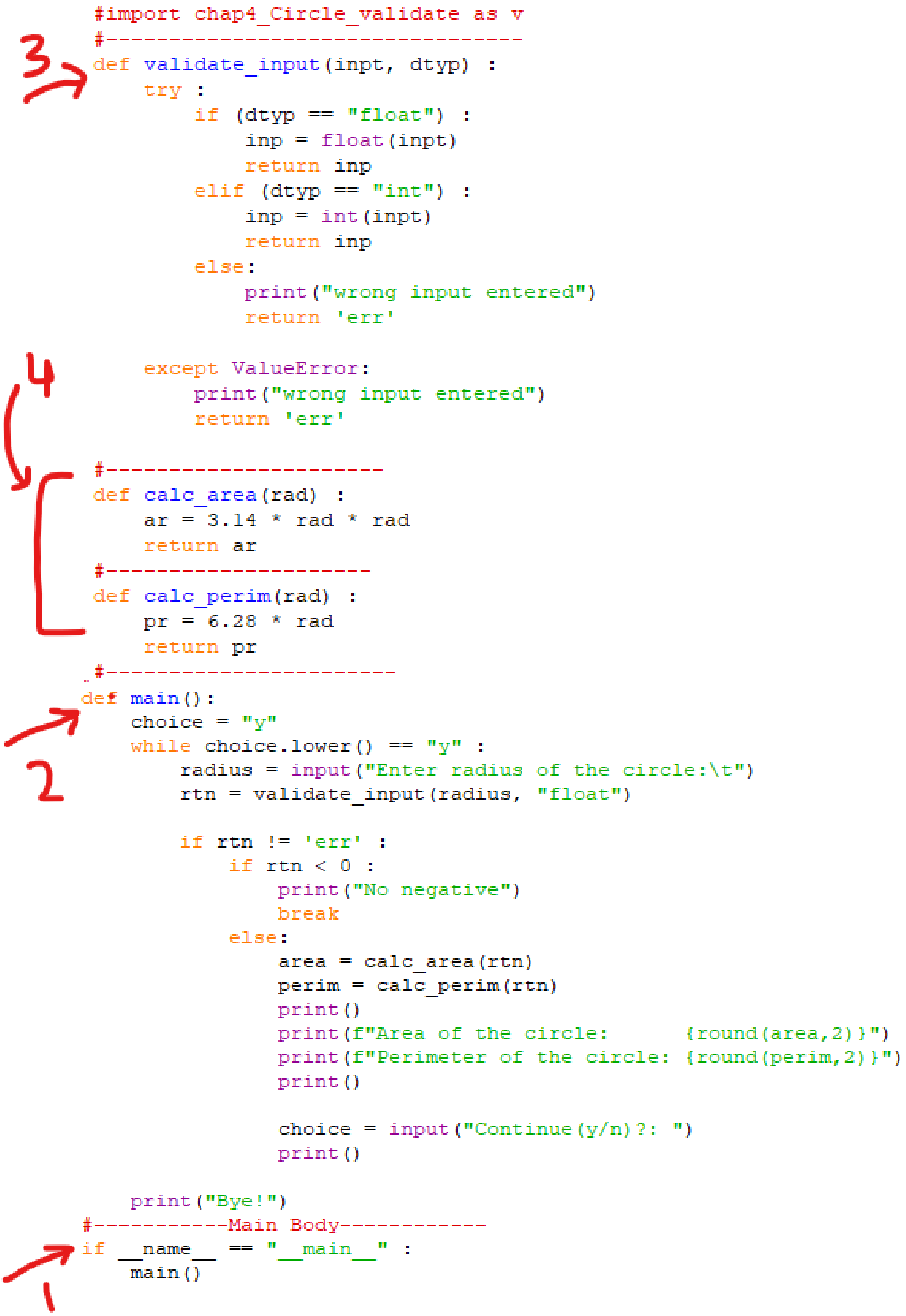
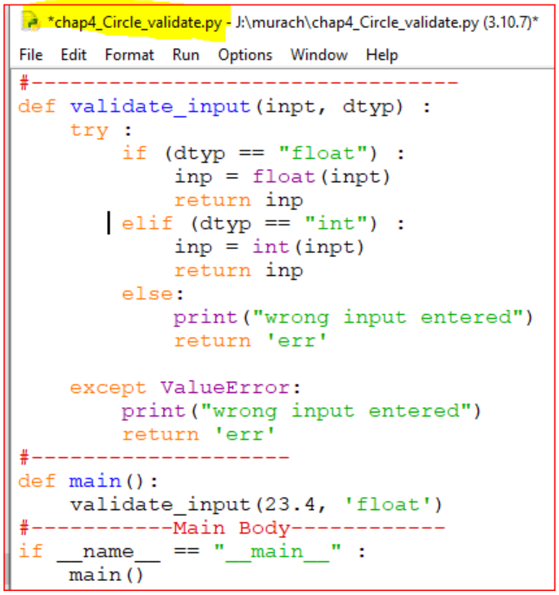


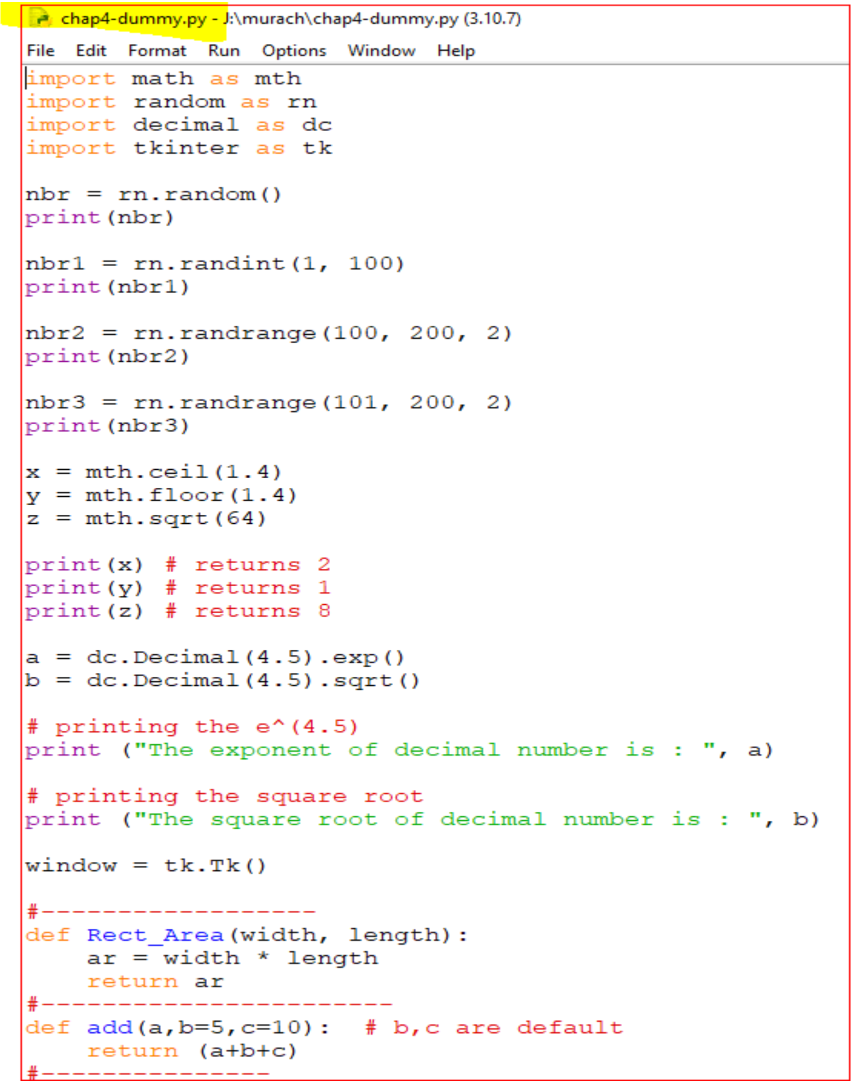
```python
#import chap4_Circle_validate as v
#----------------------------------
def validate_input(inpt, dtyp) :
    try :
        if (dtyp == "float") :
            inp = float(inpt)
            return inp
        elif  (dtyp == "int") :
            inp = int(inpt)
            return inp
        else:
            print("wrong input entered")
            return 'err'

    except ValueError:
        print("wrong input entered")
        return 'err'

#----------------------
def calc_area(rad) :
    ar = 3.14 * rad * rad
    return ar
#---------------------
def calc_perim(rad) :
    pr = 6.28 * rad
    return pr
#-----------------------
def main():
    choice = "y"
    while choice.lower() == "y" :
        radius = input("Enter radius of the circle:\t")
        rtn = validate_input(radius, "float")

        if rtn != 'err' :
            if rtn < 0 :
                print("No negative")
                break
            else:
                area = calc_area(rtn)
                perim = calc_perim(rtn)
                print()
                print(f"Area of the circle:      {round(area,2)}")
                print(f"Perimeter of the circle: {round(perim,2)}")
                print()

                choice = input("Continue(y/n)?: ")
                print()

    print("Bye!")
#-----------Main Body------------
if __name__ == "__main__" :
    main()
```

Demo showing function calls defined inline or even outside (see below) as a separate prgm

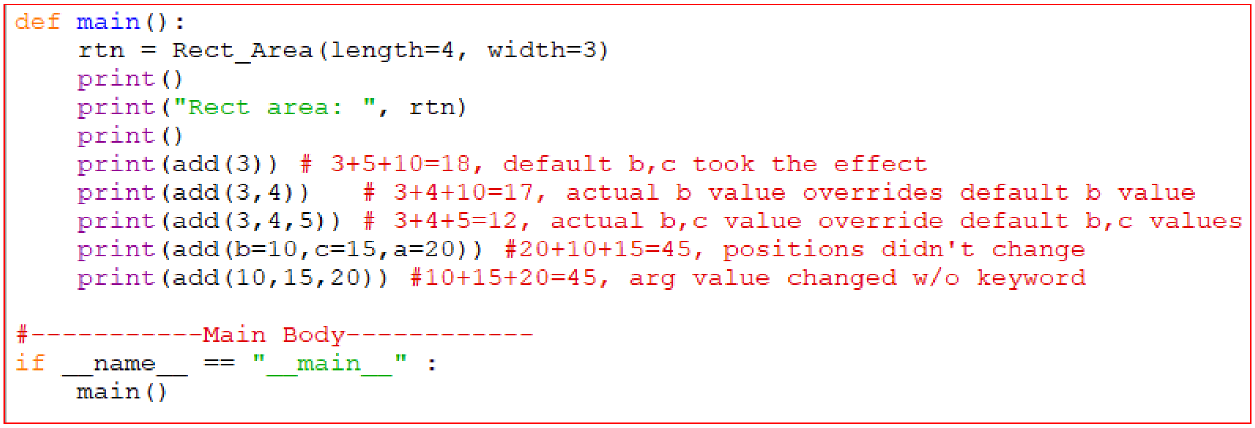

chap4_Circle_validate.py - J:\murach\chap4_Circle_validate.py (3.10.7)
File Edit Format Run Options Window Help

```
#---------------------------------
def validate_input(inpt, dtyp) :
    try :
        if (dtyp == "float") :
            inp = float(inpt)
            return inp
        elif (dtyp == "int") :
            inp = int(inpt)
            return inp
        else:
            print("wrong input entered")
            return 'err'

    except ValueError:
        print("wrong input entered")
        return 'err'
#--------------------
def main():
    validate_input(23.4, 'float')
#-----------Main Body------------
if __name__ == "__main__" :
    main()
```

←Used separately from main prgm for demo


chap4-dummy.py - J:\murach\chap4-dummy.py (3.10.7)
File Edit Format Run Options Window Help

```
import math as mth
import random as rn
import decimal as dc
import tkinter as tk

nbr = rn.random()
print(nbr)

nbr1 = rn.randint(1, 100)
print(nbr1)

nbr2 = rn.randrange(100, 200, 2)
print(nbr2)

nbr3 = rn.randrange(101, 200, 2)
print(nbr3)

x = mth.ceil(1.4)
y = mth.floor(1.4)
z = mth.sqrt(64)

print(x) # returns 2
print(y) # returns 1
print(z) # returns 8

a = dc.Decimal(4.5).exp()
b = dc.Decimal(4.5).sqrt()

# printing the e^(4.5)
print ("The exponent of decimal number is : ", a)

# printing the square root
print ("The square root of decimal number is : ", b)

window = tk.Tk()

#------------------
def Rect_Area(width, length):
    ar = width * length
    return ar
#------------------------
def add(a,b=5,c=10):  # b,c are default
    return (a+b+c)
#---------------
```



```
def main():
    rtn = Rect_Area(length=4, width=3)
    print()
    print("Rect area: ", rtn)
    print()
    print(add(3)) # 3+5+10=18, default b,c took the effect
    print(add(3,4))    # 3+4+10=17, actual b value overrides default b value
    print(add(3,4,5)) # 3+4+5=12, actual b,c value override default b,c values
    print(add(b=10,c=15,a=20)) #20+10+15=45, positions didn't change
    print(add(10,15,20)) #10+15+20=45, arg value changed w/o keyword

#-----------Main Body------------
if __name__ == "__main__" :
    main()
```

Demo showing functions with default values and also with **named arguments’** positions changed

Pseudocode for test scores avg

```
Display the welcome msg1,msg2, msg3 etc…
Initialize vars

While true
Do
        get input and store to Score variable
        if Score = ‘x’
          break
        else
          validate input with int by Calling the validate function and store into var = rtn
          if rtn = ‘err’
            print err msg “wrong input”
            break
          else
            convert score to int
            if Score between 0 & 100
              score_tot += Score
              cntr += 1
           else
              print error msg to put the score between 0 & 100
end-Do

if cntr != 0
  avg_score = Score/cntr
  print avg_score & Score

print(‘Bye’)
```