

Programming Workshop 2: Pythagoras: Pseudocode

Pythagoras 1: Simple Version: Pseudocode

Get length of a
Get length of b
Calculate a^2
Calculate b^2
Calculate c^2
Calculate square root of c^2
Output value of c

Pythagoras 2: GUI version: Pseudocode

Create GUI 400x500, title "Pythagoras"

GUI should include:

Description of program

Label and Entry box for a

Label and Entry box for b

Calculate button

Get length of a from Entry box

Get length of b from Entry box

When Calculate is clicked

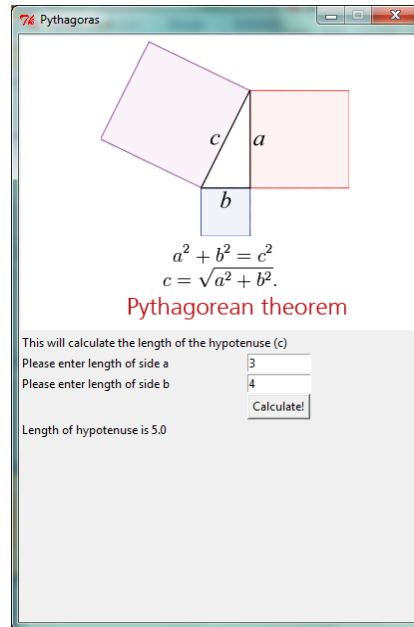
Calculate a^2

Calculate b^2

Calculate c^2

Calculate square root of c^2

Output value of c to label



Pythagoras 3: Further enhancements to GUI: Pseudocode

Calculate c when Enter is pressed

Add some validation

a and b should be integers

a and b should be greater than zero

Output validation messages using a message area

Pythagoras 4: Next Version : Pseudocode

Problem

Program is limited need a and b to work out c

What if have a and c?

Possible Solution ?

Ask first what value you need to calculate? a, b or c? Use radio button?

Then get the two known values: use three entry boxes but make unknown disabled

If calculating c: code as before

If calculating a or b: need new function to work out $\sqrt{c^2 - a^2}$ or $\sqrt{c^2 - b^2}$

Programming Workshop 2: Pythagoras: Code Snippets

Pythagoras 1: Simple Version: Code Snippets

```
import math      #for square root function
print ("This will calculate the length of the hypotenuse.")
a = input("Please enter length of side a ")
b = input("Please enter length of side b ")
asquared = int(a) * int(a)
bsquared = int(b) * int(b)
csquared = asquared + bsquared
c = math.sqrt(csquared)
print ("Length of hypotenuse is", str(c))
```

Pythagoras 2: GUI Version: Code Snippets

```
from tkinter import *      #everything there is
import math                #for square root function

def calcHypotenuse(*ignore):
    #This function takes values for a and b and calculates the value for the hypotenuse (c)
    #formula is (a*a) + (b*b) = (c*c) then need square root
    a = value_a.get()      #store values from inputs
    b = value_b.get()
    asquared = int(a) * int(a)
    bsquared = int(b) * int(b)
    csquared = asquared + bsquared
    c = math.sqrt(csquared)
    #output result
    txt = Label(app, text = "Length of hypotenuse is " + str(c))
    txt.grid(row = 6, sticky = W)

root = Tk()                # set up the GUI
root.resizable(0,0)        #stop window from being resized
app = Frame(root)          #create the frame
app.grid()                 #use a grid to layout the frame
root.title("Pythagoras")   #GUI title
root.geometry("400x600")   #GUI size
# create a Label and import the picture
picture = Label(app)
picture.grid(row = 0, columnspan = 3, sticky = W) # 'sticky = W' means left aligned (West)
pythagoras = PhotoImage(file = "pythagoras.gif") #needs to be in same directory
picture["image"] = pythagoras
# create a Label and display text
lbl = Label(app, text = "This will calculate the length of the hypotenuse (c)")
lbl.grid(row = 1, columnspan = 3, sticky = W)
# create a Label and Entry box for a
lbl = Label(app, text = "Please enter length of side a")
lbl.grid(row = 2, column = 0, sticky = W)
value_a = Entry(app, width = 10)
value_a.grid(row = 2, column = 1, sticky = W)
#also for b
lbl = Label(app, text = "Please enter length of side b")
lbl.grid(row = 3, column = 0, sticky = W)
value_b = Entry(app, width = 10)
value_b.grid(row = 3, column = 1, sticky = W)
# cursor in entry box for a
value_a.focus_force()
#create button
calc = Button(app, text = "Calculate!")
calc.grid(row = 4, column = 1, sticky = W)
calc["command"] = calcHypotenuse      #call calcHypotenuse if button clicked
```

Pythagoras 3: Further enhancements to GUI: Code Snippets

```
from tkinter import *
import math      #for square root function

# the button function
def calcHypotenuse(*ignore):
    a = value_a.get()
    b = value_b.get()
    #validation of inputs - are they integers and not zero?

    if not a.isdigit() or int(a) <= 0:
        message = "Please enter length of side a above 0"
    elif not b.isdigit() or int(b) <= 0:
        message = "Please enter length of side b above 0"
    else:      #if here we have two numbers and both above 0
        asquared = int(a) * int(a)
        bsquared = int(b) * int(b)
        csquared = asquared + bsquared
        c = math.sqrt(csquared)
        message = str(c)

    display(message)      #call function to display message

# text display function
def display(message):      #Simple text display function. Takes a string
    txt["state"] = "normal"
    txt.delete(0.0, END)
    txt.insert(0.0, message)
    txt["state"] = "disabled"

# set up the GUI
root = Tk()
root.resizable(0,0) #stop window from being resized
app = Frame(root) #create the frame
app.grid()      #use a grid to layout the frame

#bind the calcHypotenuse function to <Return> (Enter key) event
#in other words call calcHypotenuse when calc button pressed or enter pressed
root.bind('<Return>', calcHypotenuse)

[... code as version 2 ...]

calc = Button(app, text = "Calculate!")
calc["command"] = calcHypotenuse      #call calcHypotenuse if button clicked
calc.grid(row = 5, column = 1, sticky = W)

#box for messages and output
txt = Text(app, width = 49, height = 3, wrap = WORD, state = DISABLED)
txt.grid(row = 6, column = 0, columnspan = 3, padx = 1, pady = 1, sticky = W)
```

Pythagoras 4: Next Version: Code Snippets

```
from tkinter import *
import math      #for square root function

# the function called when button or Enter is pressed
def calculate(*ignore):
    rad = radio.get()    #get selected radio button state
    if rad == 3:         #calculating c (hypotenuse)
        calcHypotenuse()
    else:                #calculating a or b
        calcSide(rad)    #pass in radio button state

#calculate side
def calcSide(rad):
    a = value_a.get()
    b = value_b.get()
    c = value_c.get()
    errfound = FALSE    #flag for errors

    #validation of inputs are they integers and not zero?
    if rad == 1:         #calculating a so we have b
        if not b.isdigit() or int(b) <= 0:
            message = "Please enter length of side b above 0"
            errfound = TRUE
        else:            #calculating b so we have a
            if not a.isdigit() or int(a) <= 0:
                message = "Please enter length of side a above 0"
                errfound = TRUE

    if errfound == FALSE: #no error found yet ...
        #we'll always have c ...
        if not c.isdigit() or int(c) <= 0:
            message = "Please enter length of side c above 0"
        else:
            #if here we have two numbers and both above 0
            if rad == 1: #calculating a so we have b
                sidesquared = int(b) * int(b)
                sidetocalc = "a"    #for nice output message
            else:        #calculating b so we have a
                sidesquared = int(a) * int(a)
                sidetocalc = "b"    #for nice output message

            csquared = int(c) * int(c)
            othersidesquared = csquared - sidesquared
            otherside = math.sqrt(othersidesquared)
            message = "Length of " + sidetocalc + " is " + str(otherside)

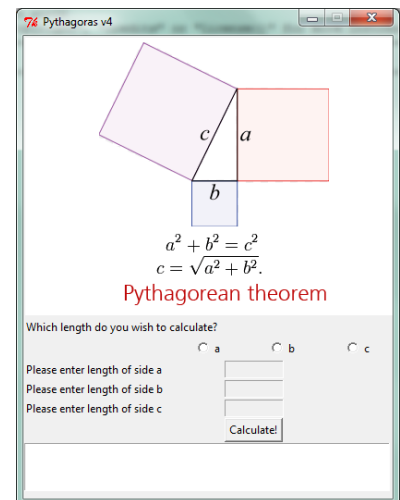
    display(message)

#calculate Hypotenuse function
def calcHypotenuse(*ignore):

    [... code as version 3 above ...]

# text display function
def display(message):

    [... code as version 3 above ...]
```



#function to enable Entry Box a, b or c depending on which radio button pressed

```
def enableEntry(*ignore):
    rad = radio.get() #get selected radio button state
    #delete contents of entries
    value_a.configure(state = "normal")
    value_b.configure(state = "normal")
    value_c.configure(state = "normal")
    value_a.delete(0, END)
    value_b.delete(0, END)
    value_c.delete(0, END)

    if rad == 1: #a selected - enable b and c
        value_a.configure(state = "disabled")
        value_b.configure(state = "normal")
        value_c.configure(state = "normal")
    elif rad == 2: #b selected - enable a and c
        value_a.configure(state = "normal")
        value_b.configure(state = "disabled")
        value_c.configure(state = "normal")
    else: #c selected - enable a and b
        value_a.configure(state = "normal")
        value_b.configure(state = "normal")
        value_c.configure(state = "disabled")

    #update all Entry boxes with new settings
    value_a.update()
    value_b.update()
    value_c.update()
```

```
root = Tk() # set up the GUI
root.resizable(0,0) #stop window from being resized
app = Frame(root) #create the frame
app.grid() #use a grid to layout the frame
```

#bind the calculate function to <Return> (Enter key) event in other words call calculate when calc button pressed or enter pressed

```
root.bind('<Return>', calculate)
```

[... code as version 3 above ...]

```
picture["image"] = pythagoras
```

```
ask_lbl = Label(app, text = "Which length do you wish to calculate?")
ask_lbl.grid(row = 1, columnspan = 3, sticky = W)
#create entry inputs - set to disabled (status changes depending on radio button)
ask_lbl = Label(app, text = "Please enter length of side a")
ask_lbl.grid(row = 3, column = 0, sticky = W)
value_a = Entry(app, width = 10, state = DISABLED)
value_a.grid(row = 3, column = 1, sticky = W)

ask_lbl = Label(app, text = "Please enter length of side b")
ask_lbl.grid(row = 4, column = 0, sticky = W)
value_b = Entry(app, width = 10, state = DISABLED)
value_b.grid(row = 4, column = 1, sticky = W)

ask_lbl = Label(app, text = "Please enter length of side c")
ask_lbl.grid(row = 5, column = 0, sticky = W)
value_c = Entry(app, width = 10, state = DISABLED)
value_c.grid(row = 5, column = 1, sticky = W)
#set up radio buttons - when selected call enableEntry function to set Entry state
radio = IntVar()
rad = Radiobutton(app, text="a", variable = radio, value = 1, command = enableEntry)
rad.grid(row = 2, column = 0, sticky = E)
rad = Radiobutton(app, text="b", variable = radio, value = 2, command = enableEntry)
rad.grid(row = 2, column = 1)
rad = Radiobutton(app, text="c", variable = radio, value = 3, command = enableEntry)
rad.grid(row = 2, column = 2, sticky = W)

calc = Button(app, text = "Calculate!")
calc["command"] = calculate #call calculate if button clicked
calc.grid(row = 6, column = 1, sticky = W)
```