

Tkinter 4

ฉัตรชัย เกษมทวีโชค

Chatchai.kase@ku.th

Import & Declare Variables

```
import tkinter as tk
import numpy as np
from tkinter import *
from tkinter import TclError, ttk
from tkinter.messagebox import showinfo, askyesno, askquestion
course_id = ['01355101','01355102','01417111','01418111','01418112','01418141','01999021','01999111','02999144']
course_details = {'01355101': ('English for Everyday Life',3),
                  '01355102': ('English for University Life',3),
                  '01417111': ('Calculus I',3),
                  '01418111': ('Introduction to Computer Science',3),
                  '01418112': ('Fundamental Programming Concepts',3),
                  '01418141': ('Intellectual Properties and Professional Ethics',3),
                  '01999021': ('Thai Language for Communication',3),
                  '01999111': ('Knowledge of the Land',2),
                  '02999144': ('Life Skill for Undergraduate Student',1)}
grades = (('A', 4.0), ('B+', 3.5), ('B', 3.0), ('C+', 2.5), ('C', 2.0),
          ('D+', 1.5), ('D', 1.0), ('F', 0.0), ('P', -1), ('NP', -2), ('W', -3))
added_course = []
added_credit = []
added_grade = []
```

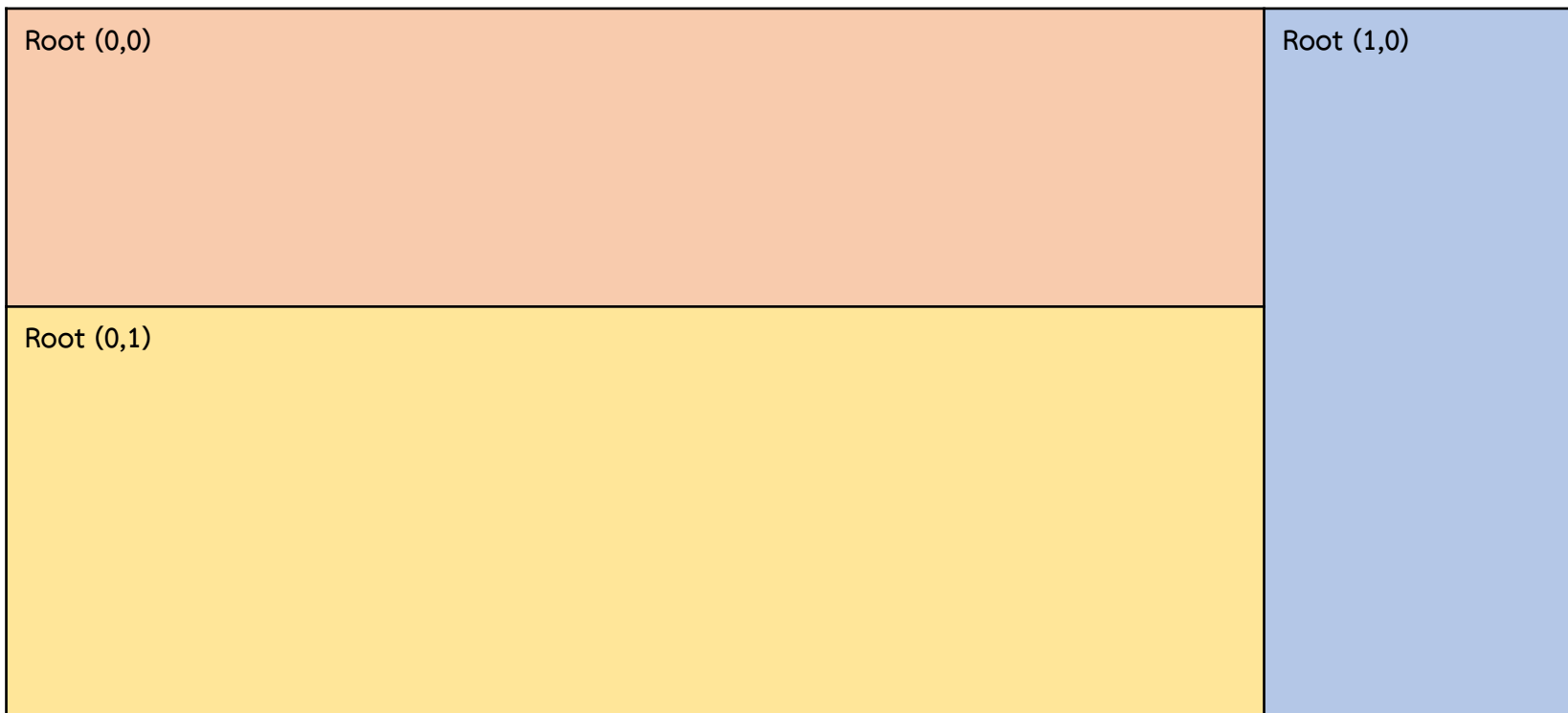
Tkinter Windows

```
root = tk.Tk()
root.title('Grade Calculator')
root.resizable(0, 0)

# สร้าง StringVar สำหรับเก็บค่าการเลือกข้อมูลรายวิชาในปัจจุบัน
selected_id = tk.StringVar()
selected_name = tk.StringVar()
selected_credit = tk.StringVar()
selected_grade = tk.StringVar()

try:
    # windows only (remove the minimize/maximize button)
    root.attributes('-toolwindow', True)
except TclError:
    print('Not supported on your platform')
```

Window Layout



```
# layout on the root window
root.columnconfigure(0, weight=4)
root.columnconfigure(1, weight=1)
root.rowconfigure(1, weight=4)
```

Replace ✕

รหัสวิชา:

ชื่อวิชา:

หน่วยกิต:

☐ A ☐ B+ ☐ B ☐ C+ ☐ C

เลือกผลการเรียน ☐ D+ ☐ D ☐ F ☐ P ☐ NP

☐ W

รหัสวิชา	ชื่อวิชา	หน่วยกิต	ผลการเรียน
----------	----------	----------	------------

Window Layout

Root (0,0)		Root (1,0)
Frame (0,0) รหัสวิชา :	Frame (1,0) <div></div>	Frame (0,1) <div>Add</div>
Frame (0,1) ชื่อวิชา :	Frame (1,1) <div></div>	Frame (0,2) <div>Delete</div>
Frame (0,2) หน่วยกิต :	Frame (1,2) <div></div>	Frame (0,3) <div>Close</div>
Frame (0,3) เลือกผลการเรียน :	Frame (1,3) <div><div><div><div><input type="radio"/> A</div><div><input type="radio"/> B+</div><div><input type="radio"/> B</div><div><input type="radio"/> C+</div><div><input type="radio"/> C</div></div><div><input type="radio"/> D+</div><div><input type="radio"/> D</div><div><input type="radio"/> F</div><div><input type="radio"/> P</div><div><input type="radio"/> NP</div></div><div><input type="radio"/> W</div></div>	

```
# layout on the root window
root.columnconfigure(0, weight=4)
root.columnconfigure(1, weight=1)
root.rowconfigure(1,weight=4)
```

Replace

รหัสวิชา:

Add

ชื่อวิชา:

Delete

หน่วยกิต:

Close

☐ A

☐ B+

☐ B

☐ C+

☐ C

☐ D+

☐ D

☐ F

☐ P

☐ NP

☐ W

เลือกผลการเรียน

จำนวนหน่วยกิต

ผลการเรียน

รหัสวิชา	ชื่อวิชา	หน่วยกิต	ผลการเรียน
----------	----------	----------	------------

Window Layout (Code)

```
input_frame, cbo_id, txt_name, txt_credit = create_input_frame(root)    # สร้าง input_frame และ widget สำหรับนำเข้าข้อมูล
input_frame.grid(column=0, row=0)

grade_table = create_grade_table(root)                                # สร้าง grade_table เป็นตารางแสดงรายการวิชา และผลการเรียน
grade_table.grid(column=0, row=1, padx=5, pady=5, sticky='nsew')

# สร้าง scrollbar สำหรับตาราง grade_table
scrollbar = ttk.Scrollbar(root, orient=tk.VERTICAL, command=grade_table.yview)
grade_table.configure(yscroll=scrollbar.set)
scrollbar.grid(row=1, column=1, sticky='nsw')

button_frame = create_button_frame(root, grade_table)                 # สร้าง button_frame และ button เพื่อการทำงานต่าง ๆ

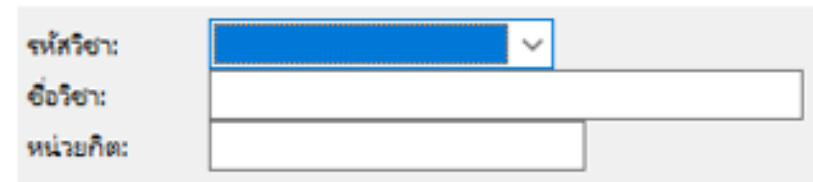
# สร้าง lbl_credits แสดงจำนวนหน่วยกิตทั้งหมด
lbl_Credits = tk.Label(button_frame, text="จำนวนหน่วยกิต", bg='green', fg='white')
lbl_Credits.grid(column=0, row=4, sticky=tk.EW, padx=5, pady=5)

# สร้าง lbl_GPA แสดงผลการเรียน
lbl_GPA = tk.Label(button_frame, text="ผลการเรียน", bg='blue', fg='white')
lbl_GPA.grid(column=0, row=5, sticky=tk.EW, padx=5, pady=5)
```

Create_Input_frame

```
def create_input_frame(container):  
    frame = ttk.Frame(container)  
    # grid layout for the input frame  
    frame.columnconfigure(0, weight=1)  
    frame.columnconfigure(1, weight=3)  
  
    # สร้าง combo box เพื่อแสดงรายการรหัสวิชาให้เลือก  
    ttk.Label(frame, text='รหัสวิชา:').grid(column=0, row=0, sticky=tk.W)  
    cbo_id = ttk.Combobox(frame, width=20, textvariable=selected_id)  
  
    cbo_id['values'] = course_id  
    cbo_id['state'] = 'readonly' # prevent typing a value  
    cbo_id.focus()  
    cbo_id.grid(column=1, row=0, sticky=tk.W)  
    cbo_id.bind('<<ComboboxSelected>>', id_changed) # bind the selected value change  
  
    # สร้าง text box เพื่อแสดงชื่อวิชา  
    ttk.Label(frame, text='ชื่อวิชา:').grid(column=0, row=1, sticky=tk.W)  
    txt_name = ttk.Entry(frame, width=40, textvariable=selected_name)  
    txt_name.grid(column=1, row=1, sticky=tk.W)  
  
    # สร้าง text box เพื่อแสดงหน่วยกิต  
    ttk.Label(frame, text='หน่วยกิต:').grid(column=0, row=2, sticky=tk.W)  
    txt_credit = ttk.Entry(frame, width=25, textvariable=selected_credit)  
    txt_credit.grid(column=1, row=2, sticky=tk.W)
```

```
def id_changed(event):  
    """ เมื่อเลือกเปลี่ยนรหัสวิชา """  
    id = selected_id.get()  
    selected_name = course_details[id][0] # เก็บชื่อวิชาใน selected_name  
    txt_name.delete(0,END)  
    txt_name.insert(0,selected_name)  
  
    selected_credit = course_details[id][1] # เก็บหน่วยกิตใน selected_credit  
    txt_credit.delete(0,END)  
    txt_credit.insert(0,selected_credit)
```



รหัสวิชา:

ชื่อวิชา:

หน่วยกิต:

Create_Input_frame

```
# สร้าง label และ frame2 แสดงรายการผลการเรียนของรายวิชา
label = ttk.Label(frame, text="เลือกผลการเรียน")
label.grid(column=0, row=3, sticky=tk.W)

frame2 = ttk.Frame(frame)
frame2.columnconfigure(0, weight=1)
frame2.columnconfigure(1, weight=1)
frame2.columnconfigure(2, weight=1)
frame2.columnconfigure(3, weight=1)
frame2.columnconfigure(4, weight=1)

# สร้างรายการตัวเลือกผลการเรียน (radio buttons) แบบ 5 คอลัมน์ต่อแถว
lr = 0
lc = i = 0
for grade in grades:
    r = ttk.Radiobutton(frame2, text=grade[0], value=grade[1], variable=selected_grade)
    r.grid(column=lc, row=lr, padx=5, pady=5, sticky=tk.W)
    i += 1
    lc = i % 5
    if lc == 0: lr += 1          # เพิ่มแถวแสดงรายการผลการเรียน เมื่อครบ 5 รายการ

frame2.grid(column=1, row=3, columnspan=2)
return frame, cbo_id, txt_name, txt_credit
```

	<input type="radio"/> A	<input type="radio"/> B+	<input type="radio"/> B	<input type="radio"/> C+	<input type="radio"/> C
เลือกผลการเรียน	<input type="radio"/> D+	<input type="radio"/> D	<input type="radio"/> F	<input type="radio"/> P	<input type="radio"/> NP
	<input type="radio"/> W				

create_grade_table (code)

```
def create_grade_table(container):  
    # กำหนดคอลัมน์ของตาราง  
    columns = ('course_id', 'course_name', 'credit', 'grade')  
    tree = ttk.Treeview(container, columns=columns, show='headings')  
  
    # สร้างหัวข้อคอลัมน์ของตาราง  
    tree.heading('course_id', text='รหัสวิชา')  
    tree.heading('course_name', text='ชื่อวิชา')  
    tree.heading('credit', text='หน่วยกิต')  
    tree.heading('grade', text='ผลการเรียน')  
  
    # กำหนดความกว้างของคอลัมน์  
    tree.column("course_id", minwidth=0, width=55, stretch=NO)  
    tree.column("course_name", minwidth=0, width=155, stretch=NO)  
    tree.column("credit", minwidth=0, width=50, stretch=NO)  
    tree.column("grade", minwidth=0, width=55, stretch=NO)  
    return tree
```

รหัสวิชา	ชื่อวิชา	หน่วยกิต	ผลการเรียน

create_button_frame (code)

```
def create_button_frame(container, table):
    frame = ttk.Frame(container)
    frame.columnconfigure(0, weight=1)

    # สร้างปุ่มการทำงาน add, delete และ close และผูกการทำงานในฟังก์ชัน click
    btn_add = ttk.Button(frame, text='Add', command=lambda: click('add', container, table)).grid(column=0, row=0)
    btn_delete = ttk.Button(frame, text='Delete', command=lambda: click('delete', container, table)).grid(column=0, row=1)
    btn_close = ttk.Button(frame, text='Close', command=lambda: click('close', container, table)).grid(column=0, row=3)

    for widget in frame.winfo_children():
        widget.grid(padx=5, pady=5)

    return frame
```

Click Add (code)

```
def click (action, root, table):
    # รวมการทำงานเมื่อผู้ใช้งานคลิกปุ่ม add, delete และ close
    if action == 'add':          # กรณีที่คลิกปุ่ม add
        course_id = selected_id.get()
        course_name = selected_name.get()
        credit = selected_credit.get()
        grade = selected_grade.get()
        if course_id == "":      # ถ้าไม่เลือกรหัสวิชา แจ้งข้อความผิดพลาด
            showinfo(title='Error', message='กรุณาเลือกรายวิชาที่ต้องการเพิ่มในตาราง')
            return
        elif grade == "":       # ถ้าไม่เลือกผลการเรียน แจ้งข้อความผิดพลาด
            showinfo(title='Error', message='กรุณาเลือกผลการเรียนของรายวิชา')
            return
        elif not course_id in added_course:  # ถ้าเลือกรหัสวิชาไม่ซ้ำ สร้างรายการใหม่ในตาราง และเพิ่มรายการวิชา หน่วยกิต และผลการเรียน
            content = [course_id, course_name, credit, grade]
            table.insert('', tk.END, values=content)
            added_course.append(course_id)
            if float(grade) >= 0.0 :
                added_credit.append(int(credit))
                added_grade.append(float(grade))
            else:
                added_credit.append(0)
                added_grade.append(0)
            updateGPA()
        else:                   # ถ้าเลือกรหัสวิชาซ้ำ แจ้งข้อความผิดพลาด
            showinfo(title='Error', message='รหัสวิชาที่ต้องการเพิ่มใหม่ซ้ำกับรหัสวิชาที่อยู่ในตารางแล้ว')
```

Click Delete & Close (code)

```
elif action == 'delete':                                # กรณีที่คลิกปุ่ม delete
    for selected_item in table.selection():
        item = table.item(selected_item)
        record = item['values']
        id = '0'+str(record[0])
        idx = added_course.index(id)
        if idx >= 0:
            answer = askyesno(title='ยืนยันการลบ',
                               message='คุณต้องการลบรายการนี้หรือไม่?')
            if answer:
                added_course.pop(idx)
                added_credit.pop(idx)
                added_grade.pop(idx)
                table.delete(selected_item)
                updateGPA()
        else:
            root.destroy()                                # กรณีที่คลิกปุ่ม close
            print('Program is closed')
```

Update_GPA

```
def updateGPA():  
    # รวมจำนวนหน่วยกิต และ GPA  
    sum_credits = np.sum(added_credit)  
    sum_grades = np.sum(np.multiply(added_credit,added_grade))  
    GPA = sum_grades / sum_credits  
  
    # แสดงค่าจำนวนหน่วยกิตและ GPA ใน lbl_Credits และ lbl_GPA  
    lbl_Credits.config(text = "จำนวนหน่วยกิต: \n" + str(sum_credits))  
    lbl_GPA.config(text="ผลการเรียน: \n" + str('{:.2f}'.format(GPA)))  
    return GPA
```