


FINALS TASK 5. Designing a Tkinter Window and adding events

PART 1. Grading PROGRAM

1. Design the window below.
2. The program should allow the user to input Prelim, Midterm, Pre Finals and Final Grade (Compute GPA by adding the Prelim, Midterms, (50% of Pre-Finals and 50% of the Final Grade) then divide by 3)
3. The user should be able to select which equivalent grade to view using Combo Box: (Letter Grade or NUMERIC GRADE)
4. Compute Button should compute the GPA and display the appropriate grade equivalent and other info in a Textarea (Text) as shown in the sample output
5. The Reset Button should clear the Radio Button Selection and the Text field entries should be cleared as well
6. The About button should display a dialog with the message: "Hello I'm your Name"

 Grading Program—□×

Input Grades

Prelim Grade: 75

Midterm Grade: 87

Pre-Final Grade: 89

Final Grade: 92

Choose an Option

Numeric Grade

▼

Transaction Summary

Prelim Grade: 75.0

Midterm Grade: 87.0

Pre-Final Grade: 89.0

Final Grade: 92.0

GPA: 84.17

Numeric Grade: 2.0

Remarks: Passed

Compute

Reset

About

Close

Grading Program

Input Grades

Prelim Grade:

a

Midterm Grade:

87

Pre-Final Grade:

89

Final Grade:

92

Choose an Option

Numeric Grade

Transaction Summary

P:

M:

P:

F:

G:

N:

R:

Compute

Reset


About

Close

Input Error

Please enter valid numeric grades!

OK

 Grading Program—□×

Input Grades

Prelim Grade: 75

Midterm Grade: 87

Pre-Final Grade: 89

Final Grade: 92

Choose an Option

Numeric Grade

▼

Transaction

P

M

P

F


G

N

R

About

×

 Hello! I'm Mark Angelo Serrano


OK

Compute

Reset

About

Close

 Grading Program—□×

Input Grades

Prelim Grade:

Midterm Grade:

Pre-Final Grade:


Final Grade:


Choose an Option

Numeric Grade

▼

Transaction Summary

 Confirm×



Are you sure you want to exit this program of mine? I'd be very sad. (But if you're ma'am, go ahead po hehe)

Yes

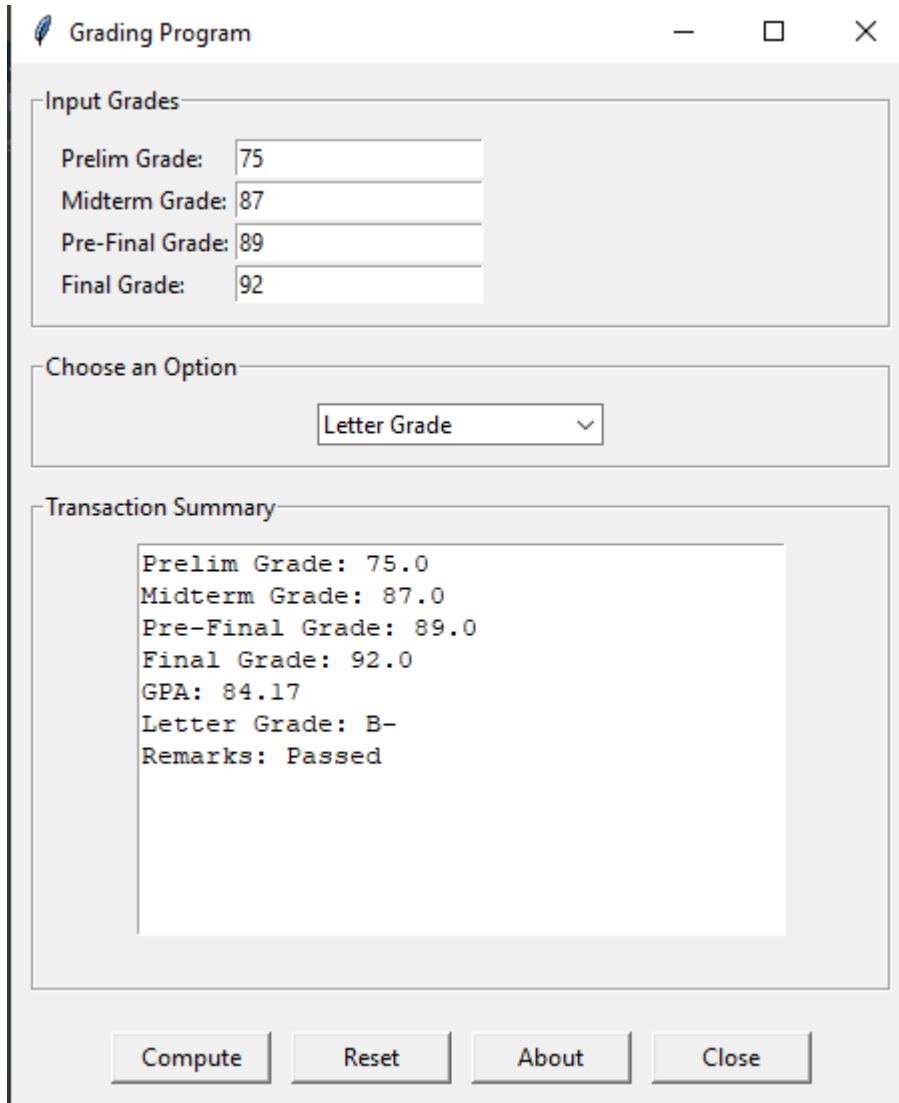
No

Compute

Reset

About

Close



A screenshot of a Java Swing window titled "Grading Program". The window has a standard title bar with a minimize button, a maximize button, and a close button. The main content area is divided into three sections. The first section, "Input Grades", contains four text input fields with labels: "Prelim Grade:" (value: 75), "Midterm Grade:" (value: 87), "Pre-Final Grade:" (value: 89), and "Final Grade:" (value: 92). The second section, "Choose an Option", contains a dropdown menu with "Letter Grade" selected. The third section, "Transaction Summary", contains a text area displaying the calculated results: "Prelim Grade: 75.0", "Midterm Grade: 87.0", "Pre-Final Grade: 89.0", "Final Grade: 92.0", "GPA: 84.17", "Letter Grade: B-", and "Remarks: Passed". At the bottom of the window, there are four buttons: "Compute", "Reset", "About", and "Close".

Grading Program

Input Grades

Prelim Grade: 75

Midterm Grade: 87

Pre-Final Grade: 89

Final Grade: 92

Choose an Option

Letter Grade

Transaction Summary

Prelim Grade: 75.0
Midterm Grade: 87.0
Pre-Final Grade: 89.0
Final Grade: 92.0
GPA: 84.17
Letter Grade: B-
Remarks: Passed

Compute Reset About Close

```
import tkinter as tk
from tkinter import ttk, messagebox

class GradingApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Grading Program")
        self.root.geometry("450x520")

        frame_input = tk.LabelFrame(root, text="Input Grades", padx=10,
pady=10)
        frame_input.pack(fill="x", padx=10, pady=10)

        tk.Label(frame_input, text="Prelim Grade:").grid(row=0, column=0,
sticky="w")
        self.entry_prelim = tk.Entry(frame_input)
        self.entry_prelim.grid(row=0, column=1)

        tk.Label(frame_input, text="Midterm Grade:").grid(row=1, column=0,
sticky="w")
        self.entry_midterm = tk.Entry(frame_input)
        self.entry_midterm.grid(row=1, column=1)

        tk.Label(frame_input, text="Pre-Final Grade:").grid(row=2, column=0,
sticky="w")
        self.entry_prefinal = tk.Entry(frame_input)
        self.entry_prefinal.grid(row=2, column=1)

        tk.Label(frame_input, text="Final Grade:").grid(row=3, column=0,
sticky="w")
        self.entry_final = tk.Entry(frame_input)
        self.entry_final.grid(row=3, column=1)

        frame_option = tk.LabelFrame(root, text="Choose an Option", padx=10,
pady=10)
        frame_option.pack(fill="x", padx=10)

        self.combo_option = ttk.Combobox(frame_option, values=["Letter
Grade", "Numeric Grade"], state="readonly")
        self.combo_option.set("Letter Grade")
        self.combo_option.pack()

        frame_output = tk.LabelFrame(root, text="Transaction Summary",
padx=10, pady=10)
        frame_output.pack(fill="both", expand=True, padx=10, pady=10)

        self.text_output = tk.Text(frame_output, height=12, width=40)
        self.text_output.pack()

        frame_buttons = tk.Frame(root)
        frame_buttons.pack(pady=10)

        tk.Button(frame_buttons, text="Compute", width=10,
command=self.compute).grid(row=0, column=0, padx=5)
        tk.Button(frame_buttons, text="Reset", width=10,
command=self.reset).grid(row=0, column=1, padx=5)
```

7OOP

```
tk.Button(frame_buttons, text="About", width=10,
command=self.about).grid(row=0, column=2, padx=5)
tk.Button(frame_buttons, text="Close", width=10,
command=self.close).grid(row=0, column=3, padx=5)

def get_letter_grade(self, gpa):
    if 96 <= gpa <= 100:
        return "A"
    elif 93 <= gpa <= 95:
        return "A-"
    elif 88 <= gpa <= 92:
        return "B"
    elif 83 <= gpa <= 87:
        return "B-"
    elif 78 <= gpa <= 82:
        return "C"
    elif 76 <= gpa <= 77:
        return "D"
    elif gpa == 75:
        return "E"
    elif 65 <= gpa <= 74:
        return "F"
    else:
        return "N/A"

def get_numeric_grade(self, gpa):
    if 97 <= gpa <= 100:
        return 1.00
    elif 94 <= gpa <= 96:
        return 1.25
    elif 90 <= gpa <= 93:
        return 1.50
    elif 87 <= gpa <= 89:
        return 1.75
    elif 84 <= gpa <= 86:
        return 2.00
    elif 81 <= gpa <= 83:
        return 2.25
    elif 78 <= gpa <= 80:
        return 2.50
    elif 76 <= gpa <= 77:
        return 2.75
    elif gpa == 75:
        return 3.00
    elif 65 <= gpa <= 74:
        return 5.00
    else:
        return "N/A"

def compute(self):
    try:
        prelim = float(self.entry_prelim.get())
        midterm = float(self.entry_midterm.get())
        prefinal = float(self.entry_prefinal.get())
        final = float(self.entry_final.get())
    except:
```



```
        messagebox.showerror("Input Error", "Please enter valid numeric
grades!")
    return

    gpa = (prelim + midterm + 0.5*prefinal + 0.5*final) / 3

    option = self.combo_option.get()
    if option == "Letter Grade":
        grade = self.get_letter_grade(gpa)
    else:
        grade = self.get_numeric_grade(gpa)

    remarks = "Passed" if gpa >= 75 else "Failed"

    self.text_output.delete("1.0", tk.END)
    self.text_output.insert(tk.END, f"Prelim Grade: {prelim}\n")
    self.text_output.insert(tk.END, f"Midterm Grade: {midterm}\n")
    self.text_output.insert(tk.END, f"Pre-Final Grade: {prefinal}\n")
    self.text_output.insert(tk.END, f"Final Grade: {final}\n")
    self.text_output.insert(tk.END, f"GPA: {gpa:.2f}\n")
    self.text_output.insert(tk.END, f"{option}: {grade}\n")
    self.text_output.insert(tk.END, f"Remarks: {remarks}\n")

    def reset(self):
        self.entry_prelim.delete(0, tk.END)
        self.entry_midterm.delete(0, tk.END)
        self.entry_prefinal.delete(0, tk.END)
        self.entry_final.delete(0, tk.END)
        self.combo_option.set("Letter Grade")
        self.text_output.delete("1.0", tk.END)

    def about(self):
        messagebox.showinfo("About", "Hello! I'm Mark Angelo Serrano")

    def close(self):
        if messagebox.askyesno("Confirm", "Are you sure you want to exit this
program of mine? I'd be very sad. (But if you're ma'am, go ahead po hehe)":
            self.root.destroy()

if __name__ == "__main__":
    root = tk.Tk()
    app = GradingApp(root)
    root.mainloop()
```