```
import tkinter as tk
import tkinter.messagebox
import time

class Time(tk.Frame):
    def __init__(self, master, *args, **kwargs):
        tk.Frame.__init__(self, master, *args, **kwargs)
        self.master = master
        self.core = False
        self.time = 0
        self.hr = 0
        self.mins = 0
        self.sec = 0
        self.Platform()

    def Platform(self):
        self.time_label = tk.Label(self, text="Enter in seconds", font=("Algerian", 15), width=20)
        self.time_label.grid(row=0,column=1)
        self.time_entry = tk.Entry(self)
        self.time_entry.grid(row=1, column=1)
        self.clock = tk.Label(self, text="00 : 00 : 00", font=("Algerian", 20), width=25)
        self.clock.grid(row=2, column=1)
        self.time_label = tk.Label(self, text="hour   min   sec", font=("Algerian", 14), width=25)
        self.time_label.grid(row=3,column=1)
        self.pause_button = tk.Button(self, text="Pause", command=lambda: self.pause())
        self.pause_button.grid(row=4,column=2)
        self.power_button = tk.Button(self, text="Start", command=lambda: self.St())
        self.power_button.grid(row=4,column=1)
        self.reset_button = tk.Button(self, text="Reset", command=lambda: self.restart())
        self.reset_button.grid(row=4,column=0)
        self.time_label = tk.Label(self, text="1 Min = 60 Sec", font=("Calibri Light", 10), width=20)
        self.time_label.grid(row=5,column=1)
        self.time_label = tk.Label(self, text="1 hr = 3600 Sec", font=("Calibri Light", 10), width=20)
```

```
    self.time_label.grid(row=6,column=1)

def Time_Converter(self):
    self.hr = self.time // 3600
    self.mins = (self.time // 60) % 60
    self.sec = self.time % 60
    return "{:02d}:{:02d}:{:02d}".format(self.hr, self.mins, self.sec)

def update(self):
    self.time = int(self.time_entry.get())
    try:
        self.clock.configure(text=self.Time_Converter())
    except:
        self.clock.configure(text="00:00:00")

def Countdown(self):
    if self.core:
        if self.time <= 0:
            self.clock.configure(text="Boom! Time out")
        else:
            self.clock.configure(text=self.Time_Converter())
            self.time -= 1
            self.after(1000, self.Countdown)

def St(self):
    try:
        self.time = int(self.time_entry.get())
        self.time_entry.delete(0, 'end')
    except:
        self.time = self.time
    self.power_button.configure(text="Stop", command=lambda: self.end())
    self.core = True
    self.Countdown()
```

```
    def end(self):
        self.power_button.configure(text="Start", command=lambda: self.St())
        self.core = False

    def restart(self):
        self.power_button.configure(text="Start", command=lambda: self.St())
        self.core = False
        self.time = 0
        self.clock["text"] = "00:00:00"

    def pause(self):
        self.pause_button.configure(text="Resume", command=lambda: self.resume())
        if self.core == True:
            self.core = False
        self.Countdown()

    def resume(self):
        self.pause_button.configure(text="Pause", command=lambda: self.pause())
        if self.core == False:
            self.core = True
        self.Countdown()

if __name__ == "__main__":
    root = tk.Tk()
    root.title("CountdownTimer")
    Time(root).pack(side="top", fill="both", expand=True)
    root.mainloop()
```

# EXPLANATION OF THE CODE

The **init** method initializes the Time class and sets up the graphical user interface for the countdown timer. It creates several labels, an entry box for the user to input the countdown time, and three buttons for starting, pausing, and resetting the timer. The “platform” method sets up the layout of the graphical user interface. It creates the labels, entry box, and buttons and sets their positions on the screen using the grid method. The Time_Converter method takes the input time in seconds and converts it to hours, minutes, and seconds. It returns a formatted string that displays the remaining time in the format "hh:mm:ss". The update method updates the countdown timer with the time entered by the user in the entry box. It calls the Time_Converte method to format the time and updates the clock label with the formatted time. The Countdown method implements the countdown functionality of the timer. It checks if the countdown has reached zero, and if not, it updates the timer label with the remaining time, decrements the time by one second, and schedules the next update using the after method. The St method starts the countdown timer. It gets the input time from the user and sets up the timer to start counting down. It changes the text of the start button to "Stop" and sets its command to end the timer when pressed. It also sets a flag to indicate that the timer is running and calls the Countdown method. The end method stops the timer. It changes the text of the start button to "Start" and sets its command to start the timer when pressed. It also sets the flag to indicate that the timer is not running. The restart method resets the timer. It changes the text of the start button to "Start" and sets its command to start the timer when pressed. It also sets the flag to indicate that the timer is not running, resets the time to zero, and updates the timer label to display "00:00:00". The pause method pauses the timer. It changes the text of the pause button to "Resume" and sets its command to resume the timer when pressed. It also sets the flag to indicate that the timer is paused and calls the Countdown method. The resume method resumes the timer. It changes the text of the pause button to "Pause" and sets its command to pause the timer when pressed. It also sets the flag to indicate that the timer is running and calls the Countdown method. Finally, the program creates an instance of the Time class and launches the graphical user interface by calling the mainloop method of the tkinter Tk class (I created the graphical user interface using the tkinter library).When the graphical user interface is running, the user can enter a time in seconds and start, pause, resume, or reset the countdown timer using the buttons provided.