import time﻿import tkinter as tk﻿class Stopwatch:﻿    def \_\_init\_\_(self, master):﻿        self.master = master﻿        self.running = False   # Indicates if the stopwatch is running﻿        self.start\_time = 0.0  # The timestamp when the stopwatch was started or resumed﻿        self.elapsed\_time = 0.0  # The total time elapsed﻿        # Create and pack a label to display the stopwatch time﻿        self.label = tk.Label(master, text="00:00:00", font=("Helvetica", 48))﻿        self.label.pack(pady=20)﻿        # Create a frame to hold the buttons﻿        button\_frame = tk.Frame(master)﻿        button\_frame.pack()﻿        # Start button﻿        self.start\_button = tk.Button(button\_frame, text="Start", command=self.start, width=10)﻿        self.start\_button.grid(row=0, column=0, padx=5)﻿        # Stop button﻿        self.stop\_button = tk.Button(button\_frame, text="Stop", command=self.stop, width=10)﻿        self.stop\_button.grid(row=0, column=1, padx=5)﻿        # Reset button﻿        self.reset\_button = tk.Button(button\_frame, text="Reset", command=self.reset, width=10)﻿        self.reset\_button.grid(row=0, column=2, padx=5)﻿        # Begin the update loop for the display﻿        self.update\_clock()﻿    def update\_clock(self):﻿        """Update the time display if the stopwatch is running."""﻿        if self.running:﻿            # Calculate elapsed time based on the current time and start time﻿            self.elapsed\_time = time.time() - self.start\_time﻿            # Break the elapsed time into hours, minutes, and seconds﻿            hours, rem = divmod(self.elapsed\_time, 3600)﻿            minutes, seconds = divmod(rem, 60)﻿            # Update the label text with a formatted string﻿            self.label.configure(text="{:02}:{:02}:{:02}".format(int(hours), int(minutes), int(seconds)))﻿        # Call this method again after 50 milliseconds﻿        self.master.after(50, self.update\_clock)﻿    def start(self):﻿        """Start or resume the stopwatch."""﻿        if not self.running:﻿            # Adjust the start time to resume from the current elapsed time﻿            self.start\_time = time.time() - self.elapsed\_time﻿            self.running = True﻿    def stop(self):﻿        """Stop the stopwatch."""﻿        self.running = False﻿    def reset(self):﻿        """Reset the stopwatch to zero."""﻿        self.running = False﻿        self.elapsed\_time = 0.0﻿        self.label.configure(text="00:00:00")﻿if \_\_name\_\_ == "\_\_main\_\_":﻿    root = [tk.Tk](tk.Tk#tk.Tk)()﻿    root.title("Stopwatch")﻿    # Optional: set a fixed size for the window﻿    root.geometry("400x200")﻿    stopwatch = Stopwatch(root)﻿    root.mainloop()