CS 1160 – Introduction To Computer Programming

Lab 10 – To-Do List

Learning Objectives

- Learn how to make programs that use classes, inheritance, and graphical user interfaces

Overview

Someone you know wants a very basic to-do list application to keep track of the tasks they have to do in the future. The requirements are that this program must keep track of the name of the task, the description of the task, the date the task is due, the time the task is due (can use a 24 hour clock i.e. 14:15 instead of 12:15 PM), whether the task is a generic/personal/school/work task.

The program must be able to remove/complete the task from the to-do list, display all upcoming tasks in the to-do list, and the program should save all the changes to the to-do list when the user quits the program without asking for a filename. When opening the to-do list program, it should load all the previously active tasks without asking the user for a filename.

All interfacing with the program must be through a GUI, meaning, you are not allowed to display anything to the user or request anything from the user inside of the IDLE interactive window or command line interface.

The Program

The program must be implemented with a GUI. Interfacing with a command line or IDLE window is not allowed. The program must have text fields where the user can enter in information for a task to be entered into a to-do list. Also, the user should be able to select whether the task is a generic task, personal task, school task, or work task. If the user selects one of these types of tasks an object with the Task/PersonalTask/SchoolTask/WorkTask class should be added to the to-do list.

When adding a new task to the to-do list, the program must check if there is a task with the same name or date+time. If there is a task that already exists with the same name or date+time, do not add it to the to-do list and warn the user. If an error occurs, such as forgetting to type a name/description/date/time for the task, inform the user.

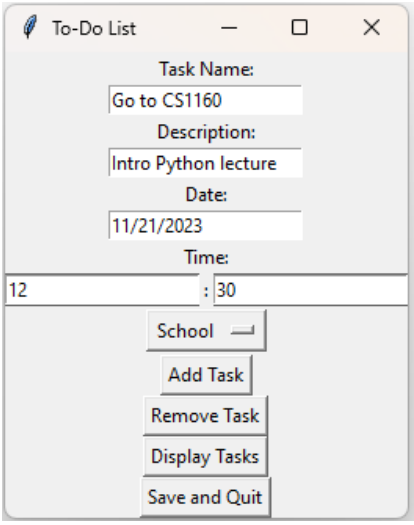
The user should be able to see all the tasks displayed in a message box when the user clicks the “Display Tasks” button. A task can be removed or “completed” by the “Remove Task” button. The “Remove Task” button should use the name that is currently in the name text field to remove that task from the to-do list. If there is no task with that name, warn the user.

If the user quits the program the entire to-do list contents must be saved to a file “todo.pickle”. Whenever the user starts the program the previous to-do list contents must be loaded from “todo.pickle”. Both saving and loading must be done without informing the user and should not display errors or success messages.

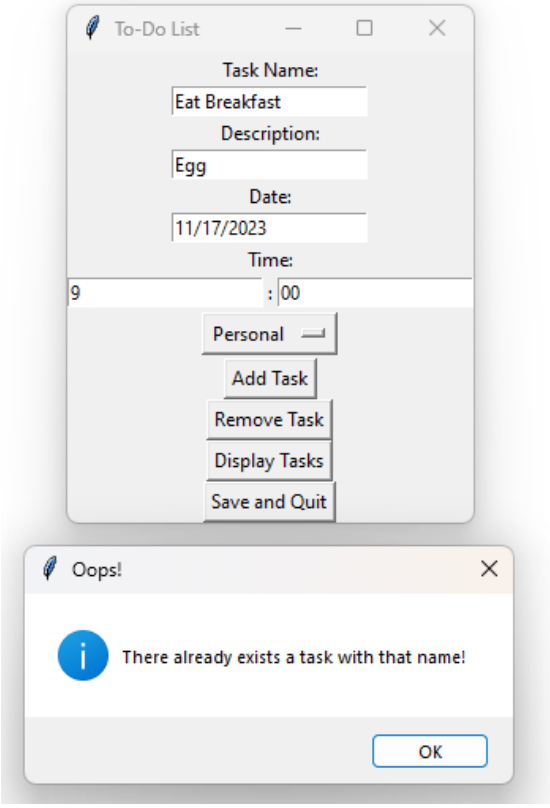
There are 5 classes required for this program:

1. ToDoApp - Contains all the GUI widgets and methods
2. TodoList - Holds the tasks and allows the ToDoApp to add/remove/display tasks. Can be saved to/loaded from a file using pickle
3. Task - Contains name, description, date, and time of task and any necessary methods. Base class for all task types
4. PersonalTask - Derived from Task, does not require any unique methods or attributes
5. SchoolTask - Derived from Task, does not require any unique methods or attributes
6. WorkTask - Derived from Task, does not require any unique methods or attributes

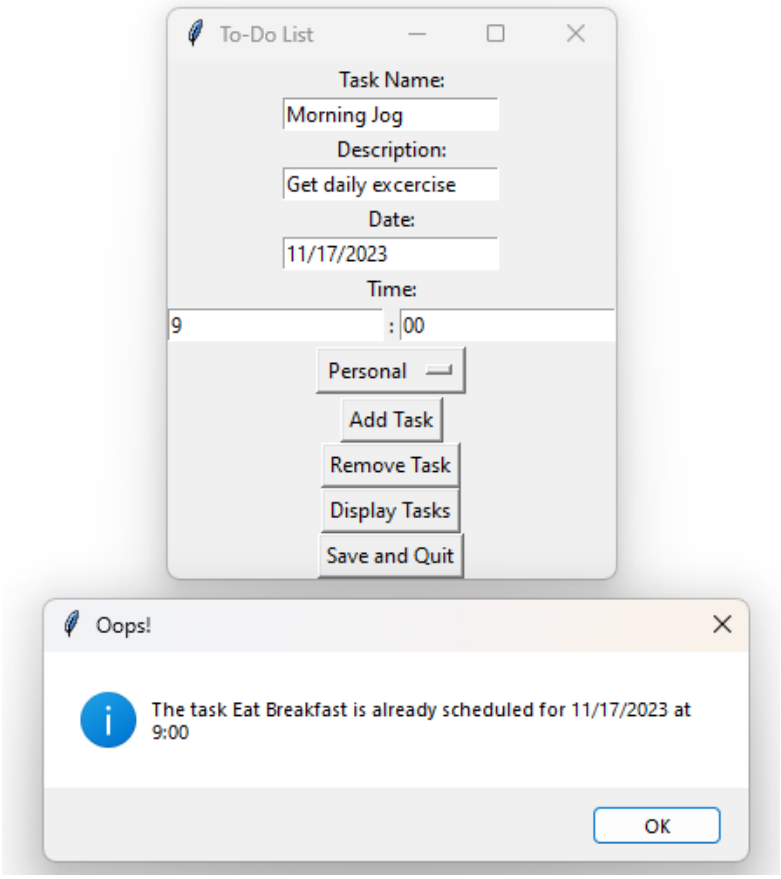
Typical View of Application:



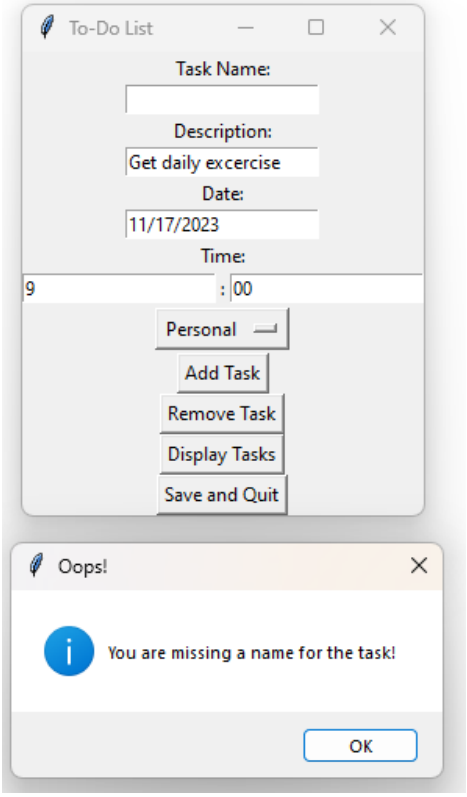
Error from entering task with same name:



Error from entering task with same date and time:



Error from entering task with missing name:



How to Submit

Save your .py Python program with your code and submit it to the drop box in Pilot.

Grading

This lab is worth 3.000 points, distributed as follows:

Task	Points
Successfully created complete GUI	0.500
Successfully adds/removes/displays tasks in to-do list with all related info	0.500
Successfully displayed any issue or message without using IDLE window	0.500
Successfully saved/loaded to-do list	0.500
Successfully took player input using input validation and without crashing	1.000
Total	3.000