HW1

To-do List!

To-do List

- Use C++ to develop a to-do list program.
- The implementation must incorporate public, private, and inheritance concepts. (-20% if not fulfilled)
- You must use pointer to save all variables in your code. (-20% if not fulfilled)
- This program should allow users to:
  - Add a task (-15% if not fulfilled)
  - View tasks (-15% if not fulfilled)
  - Edit a task (-15% if not fulfilled)
  - Delete a task (-15% if not fulfilled)

Base Class

- You must use classes that are subclasses of the basic task class shown in the provided code to store task information in the program.
- Please do not add or remove any variables from the Base Class.

```
#include <string>
```

```
class Basic_task {
protected:
    std::string *name;
    std::string *category;
    bool *completed;
};
```

Add a Task

- Enable users to add tasks and set the following attributes:
  - Task name
  - Category
  - Completed (No)
- Feel free to add custom attributes for extra features.

View Tasks

- Users can display a task list where tasks are filtered based on specific conditions, such as:
  - Category
  - Completion status

Edit and Delete a Task

- Users are allowed to edit or delete any task within a task list.
- Users can also mark a task as completed.
- Users can modify task attributes, including the task name and others.
- Users have the option to delete tasks.
- After making any deletions or edits, the updated task list should be displayed.

Optional Enhancement (25%)

- You are allowed to add additional functions and features to enhance the to-do list program, as long as the system remains functional.
- Please describe the additional features clearly in your demo video and especially in your README and paper report.
- Example:
  - Handle invalid user input.
  - In the add task function:
    - Design the interface to simplify user input, such as providing default values and intuitive input methods.
    - Allow user to set recurring task.
  - In the view tasks function:
    - Enable task display based on multiple conditions.
    - Sort tasks according to selected criteria.
  - In the edit and delete task functions:
    - Allow group edits or group deletions.
    - Support undo and redo for edits and deletions.

Report

- The report should be clear and organized.
- The report should describe how you fulfill the requirements.
- Your report must include:
  - What data structure do you store the task's information? Provide an explanation.
  - How does your code implement each operation? Provide an explanation.
  - Screenshot all the required operations.
- The file name should be named OOPDS_HW1_學號_學生名.pdf

README File

- The README file should be clear and detailed.
- The README file should describe how to use your to-do list.
- Your README file must include:
  - How to add a new task and set each attribute?
  - How to view tasks and set filtering condition?
  - How to select a specific task to edit or delete?
  - How to use the additional functions that you added?
- The file name should be named

  OOPDS_HW1_學號_學生名.txt/.md

Demo video

- You need to record all operations of the to-do list program.
- The video should be clear and demonstrate all of your working operations.
- The video should be **no more than 5 minutes**.
- Editing is allowed. (but faking your results are not allowed)
- The file name should be named OOPDS_HW1_學號_學生名.mp4

Scores and Submission

- The content will affect your grade.
- Basic score: 75
- You might get a higher score (up to 100) if you successfully enhance your system with other operations.
- Delay / Copy = 0
- Compress all your files into OOPDS_HW1_學號_學生名.zip (-20 points)
  - Report (-10 points)
  - README file (-10 points)
  - Demo video (-10 points)
  - Source code (-10 points)
- Uploaded onto E3 platform before 04/13 (Sun.) 11:55 pm.