

CSE 1321L: Programming and Problem Solving I Lab

Assignment 6

Module 6

What students will learn:

- o Design classes include attributes, constructor methods, and instance methods.
- o Creating objects of classes.
- o Implementing a collection of objects.

Content

- o Overview
- o Assignment6A: To-Do List App

Overview

For this assignment, you are going to practice implementing and designing classes defining its attributes and behaviors by implementing a constructor function and other member functions.

Final note: Do not cheat

If your temptation is to look online, don't. Come see us instead and ask questions – we are here to help. Remember, you are going to have to write code in your future job interviews, so learn it now to secure a high-paying job later.

Lastly, make sure you review the sample output and make sure the output of your program follows the exact same format including the input statements, print statement, etc. As always, user input is shown in **red** and **bold**.

Assignment 6A: To-Do List App

You are going to develop a simple To-Do List program that you may use in your day-to-day life or to organize your week or month.

To implement this, you are going to define a class for you to do list items. Furthermore, since you will have to handle multiple to do list items, such as buying groceries, going to class, studying, etc... you are going to implement a list that will contain to do list objects.

Requirements:

- o **Todo** class:
 - The constructor function will take 3 parameters. A required parameter for the title of the Todo list item, and two optional parameters for the date and time the Todo item is due.
 - The optional parameters should have a default value of ``None``
 - The constructor should initialize its attribute based on the 3 parameters.
 - The constructor should also initialize a fourth attribute called ``done`` which will be initialized as ``False``.
 - This attribute will help us know if the Todo item is done or not.
 - The class should implement the following functions:
 - `set_done()`
 - `is_done()`
 - `set_date_time()`
 - `has_due_dt()`
 - `get_todo()`
 - All the functions specified are instance functions, meaning that regardless of whether they have or not have parameters, you have to have a ``self`` reference as the first parameter.
 - **`set_done()`** should not take any parameters and will set the object attribute ``done`` to ``True``.
 - **`is_done()`** should not take any parameters and will return ``True`` or ``False`` whether the Todo object is done or not.
 - **`set_date_time()`** should take two parameters representing a date and time (both strings). This function should just re-set the object's attribute date and time.
 - **`has_due_dt()`** should not take any parameters and will return ``True`` or ``False`` whether the Todo object has both date and time attribute.
 - Remember that the Todo object sets by default date and time as ``None`` if it is constructed with just the title.
 - **`get_todo()`** should not take any parameters and will return the information regarding the Todo object.
 - If the Todo object has date and time, it should return a string with the following format (the second line has an indentation ``\t``):
 - Your todo title
Due: 00/00/00 at 00:00 AM
 - If the Todo object does not have a date and time, it should just return the title of the object.

o **Main program:**

- Initialize a list. This list will contain all your todo objects.
- Within a loop, the program should ask the user for the following options:
 - 1 - Add a To-do list item
 - 2 - Set a To-do list item
 - 3 - Set or change the due date time of a To-do item
 - 4 - Remove a todo list item due date and time
 - 5 - Print To-do list
 - 6 - Exit
- The loop should continue iterating as long the user does not enter option 6.
- If the user chooses option 1:
 - Ask if the To-do item has a due date and time.
 - Prompt the user for the title of the To-do item.
 - If the To-do item has a due date and time, prompt the user for it.
 - By calling the construction accordingly (with or without date and time), append the Todo object at the end of the list.
- If the user chooses option 2:
 - Iterate through the list and print the title of each Todo object and the corresponding index in the list.
 - Prompt the user to select which To-do item to be marked as done by inputting the corresponding index.
 - Set the corresponding object in the list as done by calling the appropriate function.
- If the user chooses option 3:
 - Iterate through the list and print the title of each Todo object and the corresponding index in the list.
 - Prompt the user to select which To-do item to set or change its due date time by inputting the corresponding index.
 - Prompt the user for a date and time.
 - Set or change the corresponding object's date and time in the list using the appropriate function.
- If the user chooses option 4:
 - Iterate through the list, only printing the title of each Todo object that has a due date and time and the corresponding index in the list.
 - Remember you have implemented a function that checks if a Todo object has a due date or not.
 - Prompt the user to select which To-do item to remove its date and time.
 - Remove the corresponding object's date and time by setting them to `None` using the appropriate function.
- If the user chooses option 5:
 - Iterate through the list and print all the unfinished Todo objects.
 - You must use the appropriate functions to check if a Todo object is done and to get the object's information.

Sample output

```
1 - Add a To-do list item
2 - Set a To-do list item as done
3 - Set or change the due date time of a To-do item
4 - Remove a To-do list item due date and time
5 - Print To-do list
6 - Exit
> 1
```

Does your To-do item have a due date and time? (y/n): **y**

What is the title: **Check lab schedule**

Enter the due date (MM/DD/YYYY): **04/06/2026**

Enter the due time (HH:MM AM/PM): **09:00 AM**

Adding your To-do list item...

```
1 - Add a To-do list item
2 - Set a To-do list item as done
3 - Set or change the due date time of a To-do item
4 - Remove a To-do list item due date and time
5 - Print To-do list
6 - Exit
> 1
```

Does your To-do item have a due date and time? (y/n): **n**

What is the title: **Work on Assignment 6**

```
1 - Add a To-do list item
2 - Set a To-do list item as done
3 - Set or change the due date time of a To-do item
4 - Remove a To-do list item due date and time
5 - Print To-do list
6 - Exit
> 1
```

Does your To-do item have a due date and time? (y/n): **y**

What is the title: **Submit Assignment 6**

Enter the due date (MM/DD/YYYY): **04/10/2026**

Enter the due time (HH:MM AM/PM): **10:00 PM**

Adding your To-do list item...

```
1 - Add a To-do list item
2 - Set a To-do list item as done
3 - Set or change the due date time of a To-do item
4 - Remove a To-do list item due date and time
5 - Print To-do list
6 - Exit
> 5
```

This is your To-do list:

- Check lab schedule
Due: 04/06/2026 at 09:00 AM
- Work on Assignment 6
- Submit Assignment 6
Due: 04/10/2026 at 10:00 PM

- 1 - Add a To-do list item
 - 2 - Set a To-do list item as done
 - 3 - Set or change the due date time of a To-do item
 - 4 - Remove a To-do list item due date and time
 - 5 - Print To-do list
 - 6 - Exit
- > 3

Choose a To-do item to set or change due date time:

- 0 Check lab schedule
- 1 Work on Assignment 6
- 2 Submit Assignment 6

> 1

Enter the due date (MM/DD/YYYY): 04/08/2026

Enter the due time (HH:MM AM/PM): 06:00 PM

- 1 - Add a To-do list item
 - 2 - Set a To-do list item as done
 - 3 - Set or change the due date time of a To-do item
 - 4 - Remove a To-do list item due date and time
 - 5 - Print To-do list
 - 6 - Exit
- > 5

This is your To-do list:

- Check lab schedule
Due: 04/06/2026 at 09:00 AM
- Work on Assignment 6
Due: 04/08/2026 at 06:00 PM
- Submit Assignment 6
Due: 04/10/2026 at 10:00 PM

- 1 - Add a To-do list item
 - 2 - Set a To-do list item as done
 - 3 - Set or change the due date time of a To-do item
 - 4 - Remove a To-do list item due date and time
 - 5 - Print To-do list
 - 6 - Exit
- > 3

Choose a To-do item to set or change due date time:

- 0 Check lab schedule
- 1 Work on Assignment 6
- 2 Submit Assignment 6

> 2

Enter the due date (MM/DD/YYYY): 04/09/2026

Enter the due time (HH:MM AM/PM): 10:00 AM

- 1 - Add a To-do list item
- 2 - Set a To-do list item as done
- 3 - Set or change the due date time of a To-do item
- 4 - Remove a To-do list item due date and time
- 5 - Print To-do list
- 6 - Exit

> 4

Choose a To-do item to remove its due date time:

- 0 Check lab schedule
- 1 Work on Assignment 6
- 2 Submit Assignment 6

> 1

- 1 - Add a To-do list item
- 2 - Set a To-do list item as done
- 3 - Set or change the due date time of a To-do item
- 4 - Remove a To-do list item due date and time
- 5 - Print To-do list
- 6 - Exit

> 4

Choose a To-do item to remove its due date time:

- 0 Check lab schedule
- 2 Submit Assignment 6

> 0

- 1 - Add a To-do list item
- 2 - Set a To-do list item as done
- 3 - Set or change the due date time of a To-do item
- 4 - Remove a To-do list item due date and time
- 5 - Print To-do list
- 6 - Exit

> 5

This is your To-do list:

- Check lab schedule
 - Work on Assignment 6
 - Submit Assignment 6
- Due: 04/09/2026 at 10:00 AM

- 1 - Add a To-do list item
- 2 - Set a To-do list item as done
- 3 - Set or change the due date time of a To-do item
- 4 - Remove a To-do list item due date and time
- 5 - Print To-do list

```
6 - Exit
> 2
```

Choose a To-do item to mark as done:

```
0 Check lab schedule
1 Work on Assignment 6
2 Submit Assignment 6
> 0
```

```
1 - Add a To-do list item
2 - Set a To-do list item as done
3 - Set or change the due date time of a To-do item
4 - Remove a To-do list item due date and time
5 - Print To-do list
6 - Exit
> 5
```

This is your To-do list:

```
- Work on Assignment 6
- Submit Assignment 6
    Due: 04/09/2026 at 10:00 AM
```

```
1 - Add a To-do list item
2 - Set a To-do list item as done
3 - Set or change the due date time of a To-do item
4 - Remove a To-do list item due date and time
5 - Print To-do list
6 - Exit
> 6
```

[Terminating program...]

Submission Instructions:

- o Programs must follow the output format provided. This includes each blank line, colons (:), and other symbols.
- o Programs must be working correctly.
- o Programs must be written in Python.
- o Programs must be submitted with the correct **.py** format.
- o Programs must be saved in files with the correct file name:
 - Assignment6A.py
- o Programs (source code files) must be uploaded to Gradescope by the due date.