Task 2

TO-DO LIST

1. Objective: Develop a console-based to-do list application.
2. Features:

- View existing tasks with their completion status.
- Add new tasks to the list.
- Mark tasks as done or undone.
- Remove tasks from the list.
- Save tasks to a file for persistence between program runs.
- Input validation to handle incorrect user choices gracefully.
- Clean and organized menu for a clear user interface.
- Encourage proper documentation of code for future reference.

Program:

```
import json
import os

# Constants
FILE_NAME = 'todo_list.json'

def load_tasks():
    """Load tasks from the file."""
    if os.path.exists(FILE_NAME):
        with open(FILE_NAME, 'r') as file:
            return json.load(file)
    return []

def save_tasks(tasks):
    """Save tasks to the file."""
```

```
    with open(FILE_NAME, 'w') as file:
        json.dump(tasks, file, indent=4)

def display_tasks(tasks):
    """Display all tasks with their completion status."""
    if not tasks:
        print("No tasks found.")
    else:
        for index, task in enumerate(tasks):
            status = "Done" if task['done'] else "Not Done"
            print(f"{index + 1}. {task['task']} - {status}")

def add_task(tasks):
    """Add a new task to the list."""
    task_description = input("Enter the task description: ").strip()
    if task_description:
        tasks.append({"task": task_description, "done": False})
        print("Task added successfully.")
    else:
        print("Task description cannot be empty.")

def mark_task(tasks, done=True):
    """Mark a task as done or undone."""
    display_tasks(tasks)
    try:
        task_index = int(input("Enter the task number to mark as done/undone: ")) - 1
        if 0 <= task_index < len(tasks):
            tasks[task_index]['done'] = done
            print("Task updated successfully.")
```

```
        else:
            print("Invalid task number.")
    except ValueError:
        print("Invalid input. Please enter a valid number.")

def remove_task(tasks):
    """Remove a task from the list."""
    display_tasks(tasks)
    try:
        task_index = int(input("Enter the task number to remove: ")) - 1
        if 0 <= task_index < len(tasks):
            tasks.pop(task_index)
            print("Task removed successfully.")
        else:
            print("Invalid task number.")
    except ValueError:
        print("Invalid input. Please enter a valid number.")

def print_menu():
    """Print the menu options."""
    print("\nTodo List Menu:")
    print("1. View Tasks")
    print("2. Add Task")
    print("3. Mark Task as Done")
    print("4. Mark Task as Not Done")
    print("5. Remove Task")
    print("6. Exit")

def main():
```

```
    """Main function to run the to-do list application."""
    tasks = load_tasks()

    while True:
        print_menu()
        choice = input("Choose an option (1-6): ").strip()

        if choice == '1':
            display_tasks(tasks)
        elif choice == '2':
            add_task(tasks)
        elif choice == '3':
            mark_task(tasks, done=True)
        elif choice == '4':
            mark_task(tasks, done=False)
        elif choice == '5':
            remove_task(tasks)
        elif choice == '6':
            save_tasks(tasks)
            print("Tasks saved. Exiting the program.")
            break
        else:
            print("Invalid choice. Please select a valid option (1-6).")

if __name__ == "__main__":
    main()
```

Output:

Todo List Menu:

1. View Tasks

2. Add Task

3. Mark Task as Done

4. Mark Task as Not Done

5. Remove Task

6. Exit

Choose an option (1-6):

1

No tasks found.

Todo List Menu:

1. View Tasks

2. Add Task

3. Mark Task as Done

4. Mark Task as Not Done

5. Remove Task

6. Exit

Choose an option (1-6):

2

Enter the task description:

coding

Task added successfully.

Todo List Menu:

1. View Tasks

2. Add Task

3. Mark Task as Done

4. Mark Task as Not Done

5. Remove Task

6. Exit

Choose an option (1-6):

2

Enter the task description:

speaking

Task added successfully.

Todo List Menu:

1. View Tasks

2. Add Task

3. Mark Task as Done

4. Mark Task as Not Done

5. Remove Task

6. Exit

Choose an option (1-6):

2

Enter the task description:

singing

Task added successfully.

Todo List Menu:

1. View Tasks

2. Add Task

3. Mark Task as Done

4. Mark Task as Not Done

5. Remove Task

6. Exit

Choose an option (1-6):

3

1. coding - Not Done

2. speaking - Not Done

3. singing - Not Done

Enter the task number to mark as done/undone:

1

Task updated successfully.

Todo List Menu:

1. View Tasks

2. Add Task

3. Mark Task as Done

4. Mark Task as Not Done

5. Remove Task

6. Exit

Choose an option (1-6):

4

1. coding - Done

2. speaking - Not Done

3. singing - Not Done

Enter the task number to mark as done/undone:

2

Task updated successfully.

Todo List Menu:

1. View Tasks

2. Add Task

3. Mark Task as Done

4. Mark Task as Not Done

5. Remove Task

6. Exit

Choose an option (1-6):

5

1. coding - Done

2. speaking - Not Done

3. singing - Not Done

Enter the task number to remove:

3

Task removed successfully.

Todo List Menu:

1. View Tasks

2. Add Task

3. Mark Task as Done

4. Mark Task as Not Done

5. Remove Task

6. Exit

Choose an option (1-6):

6

Tasks saved. Exiting the program.

=== Code Execution Successful ===