**List**

**Name: Gayatri**

**Code:**

class ToDoList:

def \_init\_(self):

self.tasks = []

def add\_task(self, task):

"""Add a new task to the list."""

self.tasks.append({"task": task, "completed": False})

print(f"Task '{task}' added.")

def remove\_task(self, task\_number):

"""Remove a task from the list."""

try:

task\_number = int(task\_number)

if task\_number > 0 and task\_number <= len(self.tasks):

task = self.tasks.pop(task\_number - 1)

print(f"Task '{task['task']}' removed.")

else:

print("Invalid task number.")

except ValueError:

print("Invalid task number.")

def mark\_completed(self, task\_number):

"""Mark a task as completed."""

try:

task\_number = int(task\_number)

if task\_number > 0 and task\_number <= len(self.tasks):

self.tasks[task\_number - 1]["completed"] = True

print(f"Task '{self.tasks[task\_number - 1]['task']}' marked as completed.")

else:

print("Invalid task number.")

except ValueError:

print("Invalid task number.")

def view\_tasks(self):

"""View all tasks in the list."""

if not self.tasks:

print("No tasks in the list.")

else:

for i, task in enumerate(self.tasks, start=1):

status = "Completed" if task["completed"] else "Not Completed"

print(f"{i}. {task['task']} - {status}")

def main():

todo\_list = ToDoList()

while True:

print("\nTo-Do List Menu:")

print("1. Add Task")

print("2. Remove Task")

print("3. Mark Task as Completed")

print("4. View Tasks")

print("5. Quit")

choice = input("Choose an option: ")

if choice == '1':

task = input("Enter a task: ")

todo\_list.add\_task(task)

elif choice == '2':

todo\_list.view\_tasks()

task\_number = input("Enter the task number to remove: ")

todo\_list.remove\_task(task\_number)

elif choice == '3':

todo\_list.view\_tasks()

task\_number = input("Enter the task number to mark as completed: ")

todo\_list.mark\_completed(task\_number)

elif choice == '4':

todo\_list.view\_tasks()

elif choice == '5':

break

else:

print("Invalid option. Please choose a valid option.")

if \_name\_ == "\_main\_":

main()