Task 1

A To-Do List application is a useful project that helps users manage

and organize their tasks efficiently. This project aims to create a

command-line or GUI-based application using Python, allowing

users to create, update, and track their to-do lists A To-Do List application is a useful project that helps users manage

and organize their tasks efficiently. This project aims to create a

command-line or GUI-based application using Python, allowing

users to create, update, and track their to-do lists.

Code:-

import pickle

import os

def \_init\_(self, title, completed=False):

self.title = title

self.completed = completed

def \_repr\_(self):

return f"{'[X]' if self.completed else '[ ]'} {self.title}"

def \_init\_(self, filename='todo\_list.pkl'):

self.filename = filename

self.tasks = self.load\_tasks()

def add\_task(self, title):

task = Task(title)

self.tasks.append(task)

self.save\_tasks()

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

def view\_tasks(self):

if not self.tasks:

print("Your to-do list is empty!")

else:

print("Your to-do list:")

for index, task in enumerate(self.tasks, 1):

print(f"{index}. {task}")

def update\_task(self, task\_number, new\_title):

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

self.tasks[task\_number - 1].title = new\_title

self.save\_tasks()

print(f"Task {task\_number} updated to '{new\_title}'.")

else:

print("Invalid task number!")

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

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

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

self.save\_tasks()

print(f"Task '{removed\_task.title}' deleted.")

else:

print("Invalid task number!")

def mark\_task\_complete(self, task\_number):

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

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

self.save\_tasks()

print(f"Task {task\_number} marked as complete.")

else:

print("Invalid task number!")

def save\_tasks(self):

with open(self.filename, 'wb') as file:

pickle.dump(self.tasks, file)

def load\_tasks(self):

if os.path.exists(self.filename):

with open(self.filename, 'rb') as file:

return pickle.load(file)

return []

def main():

todo\_list = ToDoList()

while True:

print("\nTo-Do List Application")

print("1. View tasks")

print("2. Add task")

print("3. Update task")

print("4. Delete task")

print("5. Mark task as complete")

print("6. Exit")

choice = input("Choose an option: ")

if choice == '1':

todo\_list.view\_tasks()

elif choice == '2':

title = input("Enter task title: ")

todo\_list.add\_task(title)

elif choice == '3':

task\_number = int(input("Enter task number to update: "))

new\_title = input("Enter new task title: ")

todo\_list.update\_task(task\_number, new\_title)

elif choice == '4':

task\_number = int(input("Enter task number to delete: "))

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

elif choice == '5':

task\_number = int(input("Enter task number to mark as complete: "))

todo\_list.mark\_task\_complete(task\_number)

elif choice == '6':

print("Exiting the application.")

break

else:

print("Invalid choice! Please try again.")

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

main()

Output:-

