class CleanupTask:

def \_init\_(self, task\_id, description, location, assigned\_to=None):

self.task\_id = task\_id

self.description = description

self.location = location

self.assigned\_to = assigned\_to

self.completed = False

def mark\_complete(self):

self.completed = True

def \_repr\_(self):

return f"Task({self.task\_id}, {self.description}, {self.location}, Assigned to: {self.assigned\_to}, Completed: {self.completed})"

class CleanupCoordinator:

def \_init\_(self):

self.tasks = {} # Using a dictionary to store tasks by their IDs

def add\_task(self, task):

self.tasks[task.task\_id] = task

def get\_task(self, task\_id):

return self.tasks.get(task\_id)

def update\_task(self, task\_id, description=None, location=None, assigned\_to=None):

task = self.get\_task(task\_id)

if task:

if description:

task.description = description

if location:

task.location = location

if assigned\_to:

task.assigned\_to = assigned\_to

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

if task\_id in self.tasks:

del self.tasks[task\_id]

def list\_tasks(self):

return list(self.tasks.values())

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

task = self.get\_task(task\_id)

if task:

task.mark\_complete()

def track\_cleanup(self):

completed\_tasks = [task for task in self.tasks.values() if task.completed]

pending\_tasks = [task for task in self.tasks.values() if not task.completed]

return {

"completed\_tasks": completed\_tasks,

"pending\_tasks": pending\_tasks

}

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

coordinator = CleanupCoordinator()

# Creating tasks

task1 = CleanupTask(task\_id=1, description="Pick up trash", location="Stadium Entrance")

task2 = CleanupTask(task\_id=2, description="Clean restrooms", location="West Wing")

# Adding tasks

coordinator.add\_task(task1)

coordinator.add\_task(task2)

# Listing tasks

print("All Tasks:")

for task in coordinator.list\_tasks():

print(task)

# Updating a task

coordinator.update\_task(1, assigned\_to="John Doe")

coordinator.update\_task(2, assigned\_to="Karna")

# Marking a task complete

coordinator.mark\_task\_complete(1)

coordinator.mark\_task\_complete(2)

# Tracking cleanup status

status = coordinator.track\_cleanup()

print("\nCleanup Status:")

print("Completed Tasks:", status["completed\_tasks"])

print("Pending Tasks:", status["pending\_tasks"])

# Deleting a task

coordinator.delete\_task(1)

coordinator.delete\_task(2)

while True:

choice = int(input("Enter your choice:"))

print("Enter 1 to add task")

print("Enter 2 to update task")

print("Enter 3 to mark as complete")

if choice == 1:

task1 = CleanupTask(task\_id=1, description="Pick up trash", location="Stadium Entrance")

elif choice == 2:

coordinator.update\_task(1, assigned\_to="John Doe")

elif choice == 3:

coordinator.mark\_task\_complete(1)

print("\nTasks after deletion:")

print(coordinator.list\_tasks())