# PROGRAM:

# class Schedule:

# def \_init\_(self, schedule\_id, project\_name, start\_date, end\_date):

# self.schedule\_id = schedule\_id

# self.project\_name = project\_name

# self.start\_date = start\_date

# self.end\_date = end\_date

# 

# def \_repr\_(self):

# return f"Schedule(id={self.schedule\_id}, project={self.project\_name}, start={self.start\_date}, end={self.end\_date})“

# class ProjectTimeline:

# def \_init\_(self, timeline\_id):

# self.timeline\_id = timeline\_id

# self.schedules = {}

# def add\_schedule(self, schedule):

# if schedule.schedule\_id in self.schedules:

# raise ValueError(f"Schedule ID {schedule.schedule\_id} already exists.") self.schedules[schedule.schedule\_id] = schedule

# def get\_schedule(self, schedule\_id):

# return self.schedules.get(schedule\_id, None)

# def update\_schedule(self, schedule\_id, start\_date=None, end\_date=None):

# schedule = self.get\_schedule(schedule\_id)

# if not schedule:

# raise ValueError("Schedule not found.")

# if start\_date:

# schedule.start\_date = start\_date

# if end\_date:

# schedule.end\_date = end\_date

# def delete\_schedule(self, schedule\_id):

# if schedule\_id not in self.schedules:

# raise ValueError("Schedule not found.")

# del self.schedules[schedule\_id]

# def \_repr\_(self):

# return f"ProjectTimeline(id={self.timeline\_id}, schedules={self.schedules})“

# class ProjectScheduler:

# def \_init\_(self):

# self.timelines = {}

def create\_timeline(self, timeline\_id):

if timeline\_id in self.timelines:

raise ValueError(f"Timeline ID {timeline\_id} already exists.") self.timelines[timeline\_id] = ProjectTimeline(timeline\_id)

def get\_timeline(self, timeline\_id):

return self.timelines.get(timeline\_id, None)

def adjust\_schedule(self, timeline\_id, schedule\_id, start\_date=None, end\_date=None):

timeline = self.get\_timeline(timeline\_id)

if not timeline:

raise ValueError("Timeline not found.") timeline.update\_schedule(schedule\_id, start\_date, end\_date)

def \_repr\_(self):

return f"ProjectScheduler(timelines={self.timelines})"

# # Example usage

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

# scheduler = ProjectScheduler()

# # Create a project timeline

# scheduler.create\_timeline("timeline\_1")

# 

# # Create a schedule and add it to the timeline

# schedule1 = Schedule(schedule\_id="schedule\_1", project\_name="Project A", start\_date="2024-01-01", end\_date="2024-01-31") scheduler.get\_timeline("timeline\_1").add\_schedule(schedule1)

# # Retrieve and print the timeline

# print(scheduler.get\_timeline("timeline\_1"))

# 

# # Adjust the schedule

# scheduler.adjust\_schedule("timeline\_1", "schedule\_1", end\_date="2024-02-15”)

# Retrieve and print the updated timeline print(scheduler.get\_timeline("timeline\_1"))

# Delete the schedule scheduler.get\_timeline("timeline\_1").delete\_schedule("schedule\_1")

# Print the timeline after deletion

print(scheduler.get\_timeline("timeline\_1"))

***Output*** :

* ProjectScheduler(Timelines: [ProjectTimeline(ID: Timeline\_1, Schedules: [Schedule(ID: Schedule\_1, Name: Task 1, Start: 2024-09-01, End: 2024-09-05), Schedule(ID: Schedule\_2, Name: Task 2, Start: 2024-09-06, End: 2024-09-10)])])
* Adjusted Schedule: Schedule(ID: Schedule\_1, Name: Task 1, Start: 2024-09-01, End: 2024-09-06)
* Deleted Schedule: Schedule(ID: Schedule\_2, Name: Task 2, Start: 2024-09-06, End: 2024-09-10)
* ProjectScheduler(Timelines: [ProjectTimeline(ID: Timeline\_1, Schedules: [Schedule(ID: Schedule\_1, Name: Task 1, Start: 2024-09-01, End: 2024-09-06)])])