import random

from datetime import datetime, timedelta

class Team:

def \_init\_(self, name):

self.name = name

class Match:

def \_init\_(self, team1, team2, date):

self.team1 = team1

self.team2 = team2

self.date = date

class Tournament:

def \_init\_(self, name):

self.name = name

self.teams = []

self.matches = []

def add\_team(self, team):

self.teams.append(team)

def remove\_team(self, team):

self.teams.remove(team)

def get\_teams(self):

return self.teams

def schedule\_match(self, team1, team2, date):

match = Match(team1, team2, date)

self.matches.append(match)

def get\_matches(self):

return self.matches

def automate\_match\_scheduling(self, start\_date, num\_days):

for i in range(len(self.teams)):

for j in range(i + 1, len(self.teams)):

match\_date = start\_date + timedelta(days=random.randint(0, num\_days))

self.schedule\_match(self.teams[i], self.teams[j], match\_date)

def generate\_tournament\_brackets(self):

if len(self.teams) < 2:

return "Not enough teams to generate brackets."

bracket = []

random.shuffle(self.teams)

while len(self.teams) > 1:

round\_matches = []

while len(self.teams) >= 2:

team1 = self.teams.pop()

team2 = self.teams.pop()

round\_matches.append((team1.name, team2.name))

bracket.append(round\_matches)

return bracket

# Example Usage

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

tournament = Tournament("Champions League")

tournament.add\_team(Team("CSK"))

tournament.add\_team(Team("RCB"))

tournament.add\_team(Team("MI"))

tournament.add\_team(Team("DEHLI CAP"))

# Schedule matches automatically

start\_date = datetime.now()

tournament.automate\_match\_scheduling(start\_date, 30)

# Generate tournament brackets

brackets = tournament.generate\_tournament\_brackets()

print("Scheduled Matches:")

for match in tournament.get\_matches():

print(f"{match.team1.name} vs {match.team2.name} on {match.date}")

print("\nTournament Brackets:")

for round in brackets:

for match in round:

print(f"{match[0]} vs {match[1]}")