# Athlete Sponsorship Management System

class Athlete:

def \_init\_(self, name, sport, country):

self.name = name

self.sport = sport

self.country = country

self.sponsorships = [] # List to hold athlete's sponsorships

def add\_sponsorship(self, sponsorship):

self.sponsorships.append(sponsorship)

def update\_sponsorship(self, company\_name, amount=None, duration=None, benefits=None):

for sponsorship in self.sponsorships:

if sponsorship.company == company\_name:

sponsorship.update\_sponsorship(amount, duration, benefits)

def remove\_sponsorship(self, company\_name):

self.sponsorships = [s for s in self.sponsorships if s.company != company\_name]

def get\_total\_benefits(self):

return sum([sponsorship.benefits for sponsorship in self.sponsorships])

def display\_info(self):

info = f"Athlete Name: {self.name}\nSport: {self.sport}\nCountry: {self.country}\n"

info += "Sponsorships:\n"

if self.sponsorships:

for sponsorship in self.sponsorships:

info += sponsorship.display\_info() + "\n"

else:

info += " No Sponsorships\n"

info += f"Total Benefits Received: ${self.get\_total\_benefits()}\n"

return info

class Sponsorship:

def \_init\_(self, company, amount, duration, benefits):

self.company = company

self.amount = amount

self.duration = duration

self.benefits = benefits # Benefits received by the athlete from the sponsorship

def update\_sponsorship(self, amount=None, duration=None, benefits=None):

if amount is not None:

self.amount = amount

if duration is not None:

self.duration = duration

if benefits is not None:

self.benefits = benefits

def display\_info(self):

return f"Company: {self.company}, Amount: ${self.amount}, Duration: {self.duration} years, Benefits Received: ${self.benefits}"

class SponsorshipManager:

def \_init\_(self):

self.athletes = [] # List to hold all athletes

# CRUD operations for athletes

def add\_athlete(self, name, sport, country):

athlete = Athlete(name, sport, country)

self.athletes.append(athlete)

def find\_athlete(self, name):

for athlete in self.athletes:

if athlete.name == name:

return athlete

return None

def update\_athlete(self, name, new\_name=None, new\_sport=None, new\_country=None):

athlete = self.find\_athlete(name)

if athlete:

if new\_name:

athlete.name = new\_name

if new\_sport:

athlete.sport = new\_sport

if new\_country:

athlete.country = new\_country

else:

print(f"Athlete {name} not found.")

def remove\_athlete(self, name):

self.athletes = [athlete for athlete in self.athletes if athlete.name != name]

# CRUD operations for sponsorships

def add\_sponsorship\_to\_athlete(self, athlete\_name, company, amount, duration, benefits):

athlete = self.find\_athlete(athlete\_name)

if athlete:

sponsorship = Sponsorship(company, amount, duration, benefits)

athlete.add\_sponsorship(sponsorship)

else:

print(f"Athlete {athlete\_name} not found.")

def update\_sponsorship\_for\_athlete(self, athlete\_name, company, amount=None, duration=None, benefits=None):

athlete = self.find\_athlete(athlete\_name)

if athlete:

athlete.update\_sponsorship(company, amount, duration, benefits)

else:

print(f"Athlete {athlete\_name} not found.")

def remove\_sponsorship\_from\_athlete(self, athlete\_name, company):

athlete = self.find\_athlete(athlete\_name)

if athlete:

athlete.remove\_sponsorship(company)

else:

print(f"Athlete {athlete\_name} not found.")

# Display athlete and sponsorship information

def display\_athlete\_info(self, athlete\_name):

athlete = self.find\_athlete(athlete\_name)

if athlete:

print(athlete.display\_info())

else:

print(f"Athlete {athlete\_name} not found.")

def display\_all\_athletes(self):

if self.athletes:

for athlete in self.athletes:

print(athlete.display\_info())

else:

print("No athletes in the system.")

# Example usage

manager = SponsorshipManager()

# Adding athletes

manager.add\_athlete("Virat kohli", "cricket", "India")

manager.add\_athlete("Jane Smith", "Tennis", "UK")

# Adding sponsorships

manager.add\_sponsorship\_to\_athlete("Virat kohli", "Nike", 500000, 3, 150000)

manager.add\_sponsorship\_to\_athlete("Jane Smith", "Adidas", 300000, 2, 100000)

# Display athlete info

manager.display\_athlete\_info("Virat kohli")

manager.display\_athlete\_info("Jane Smith")

# Update sponsorship for Virat kohli

manager.update\_sponsorship\_for\_athlete("Virat kohli", "Nike", benefits=200000)

# Display updated info

manager.display\_athlete\_info("Virat kohli")

# Remove a sponsorship

manager.remove\_sponsorship\_from\_athlete("Virat kohli", "Nike")

manager.remove\_sponsorship\_from\_athlete("Jane Smith","Adidas")

# Display updated info after removal

manager.display\_athlete\_info("Virat kohli")

manager.display\_athlete\_info("Jane Smith")

# Display all athletes

manager.display\_all\_athletes()

Output:

Athlete Name: Virat kohli

Sport: cricket

Country: India

Sponsorships:

Company: Nike, Amount: $500000, Duration: 3 years, Benefits Received: $150000

Total Benefits Received: $150000

Athlete Name: Jane Smith

Sport: Tennis

Country: UK

Sponsorships:

Company: Adidas, Amount: $300000, Duration: 2 years, Benefits Received: $100000

Total Benefits Received: $100000

Athlete Name: Virat kohli

Sport: cricket

Country: India

Sponsorships:

Company: Nike, Amount: $500000, Duration: 3 years, Benefits Received: $200000

Total Benefits Received: $200000

Athlete Name: Virat kohli

Sport: cricket

Country: India

Sponsorships:

No Sponsorships

Total Benefits Received: $0

Athlete Name: Jane Smith

Sport: Tennis

Country: UK

Sponsorships:

No Sponsorships

Total Benefits Received: $0

Athlete Name: Virat kohli

Sport: cricket

Country: India

Sponsorships:

No Sponsorships

Total Benefits Received: $0

Athlete Name: Jane Smith

Sport: Tennis

Country: UK

Sponsorships:

No Sponsorships

Total Benefits Received: $0