Here's a Python program fulfilling your requirements:

Fitness Tracker Program

class FitnessTracker:

def \_\_init\_\_(self):

self.activities = []

def add\_activity(self):

"""Add new fitness activity."""

name = input("Enter activity name: ")

while True:

try:

duration = float(input("Enter duration (hours): "))

if duration <= 0:

print("Duration must be positive.")

else:

break

except ValueError:

print("Invalid duration. Please enter a number.")

while True:

try:

calories = float(input("Enter calories burned: "))

if calories < 0:

print("Calories burned cannot be negative.")

else:

break

except ValueError:

print("Invalid calories. Please enter a number.")

self.activities.append({"name": name, "duration": duration, "calories": calories})

print("Activity added successfully!")

def update\_activity(self):

"""Update existing activity details."""

if not self.activities:

print("No activities to update.")

return

print("Activities:")

for i, activity in enumerate(self.activities, 1):

print(f"{i}. {activity['name']}")

while True:

try:

choice = int(input("Enter activity number to update: "))

if 1 <= choice <= len(self.activities):

break

else:

print("Invalid choice.")

except ValueError:

print("Invalid choice. Please enter a number.")

activity = self.activities[choice - 1]

print("Enter new details (press Enter to skip):")

activity["name"] = input(f"Name ({activity['name']}): ") or activity["name"]

while True:

new\_duration = input(f"Duration ({activity['duration']} hours): ")

if not new\_duration:

break

try:

activity["duration"] = float(new\_duration)

if activity["duration"] <= 0:

print("Duration must be positive.")

else:

break

except ValueError:

print("Invalid duration. Please enter a number.")

while True:

new\_calories = input(f"Calories ({activity['calories']}): ")

if not new\_calories:

break

try:

activity["calories"] = float(new\_calories)

if activity["calories"] < 0:

print("Calories burned cannot be negative.")

else:

break

except ValueError:

print("Invalid calories. Please enter a number.")

print("Activity updated successfully!")

def calculate\_calories(self):

"""Calculate total calories burned."""

total\_calories = sum(activity["calories"] for activity in self.activities)

print(f"Total calories burned: {total\_calories:.2f}")

def generate\_report(self):

"""Generate report sorted by duration."""

if not self.activities:

print("No activities to report.")

return

sorted\_activities = sorted(self.activities, key=lambda x: x["duration"], reverse=True)

print("Fitness Activity Report (Sorted by Duration):")

for i, activity in enumerate(sorted\_activities, 1):

print(f"{i}. {activity['name']}: {activity['duration']:.2f} hours, {activity['calories']:.2f} calories")

def main():

tracker = FitnessTracker()

while True:

print("\nFitness Tracker Menu:")

print("1. Add Activity")

print("2. Update Activity")

print("3. Calculate Total Calories")

print("4. Generate Report")

print("5. Exit")

while True:

try:

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

if 1 <= choice <= 5:

break

else:

print("Invalid choice. Please choose again.")

except ValueError:

print("Invalid choice. Please enter a number.")

if choice == 1:

tracker.add\_activity()

elif choice == 2:

tracker.update\_activity()

elif choice == 3:

tracker.calculate\_calories()

elif choice == 4:

tracker.generate\_report()

elif choice == 5:

print("Exiting program. Goodbye!")

break

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

main()