

# Fitness App

Create a console application to help users manage workouts, exercises, and fitness progress. Users can create and organize workout plans, add exercises with sets, track calories burned, log completed workouts, mark favorites, and save/load fitness data between sessions.

## Class: Exercise

Represents a single exercise.

### ATTRIBUTES:

- string name – exercise name
- string muscleGroup – muscle group being targeted (Chest, Legs, Back, etc)
- string type – type of exercise (Cardio, Strength, Flexibility, Balance)
- int caloriesBurned – estimated calories burned per session

### METHODS:

- void display() - display exercise details

## Class: Set

Represents one set of an exercise.

### ATTRIBUTES:

- int reps – number of repetitions

\*constructors, destructors, and getters/setters are **NOT** explicitly listed in **METHODS** but should be implemented and used where needed.

- double weight – weight used (kg or lbs)
- int duration – duration in seconds (for cardio)

#### **METHODS:**

- void display() - display set details

## **Class: Workout**

Represents a complete workout session consisting of exercises & sets.

#### **ATTRIBUTES:**

- string title – workout name
- string category – Upper Body, Lower Body, Full Body, Cardio
- int totalTime – work duration in minutes
- vector<Exercise\*> exercises – exercises included
- vector<vector<Set\*>> sets – sets per exercise

Tip: sets[i] refers to the list of sets for exercise[i], sets[i][j] returns the j-th Set\* for exercise[i]

- bool isFavorite – favorite status

#### **METHODS:**

- void addExercise() - prompts user to add an exercise
- void addSet(int exerciseIndex) - prompts user to add a set
- void toggleFavorite() - unfavorites/favorites a workout
- int calculateCalories() - calculates calories for a workout
- void display() - displays details about the workout

\*constructors, destructors, and getters/setters are **NOT** explicitly listed in **METHODS** but should be implemented and used where needed.

## Class: Log

Stores completed workout sessions.

### ATTRIBUTES:

- string date – date a workout was performed
- Workout\* workoutPerformed – pointer to a Workout
- string notes – notes about a workout

### METHODS:

- void display() - displays information about a log

## Class: App

Manages all workouts and logs.

### ATTRIBUTES:

vector<Workout\*> workouts – vector of pointers to workouts

vector<Log\*> history – vector of pointers to logged workouts

### METHODS:

void createWorkout()

void viewAllWorkouts()

void viewByCategory(string category)

void viewFavorites()

\*constructors, destructors, and getters/setters are **NOT** explicitly listed in **METHODS** but should be implemented and used where needed.

void logWorkout()

void deleteWorkout(int index)

void saveToFile(ofstream& output) - saves app info to a file

void loadFromFile(ifstream& input) - loads app info from a file

## Class: User

Represents the app user.

### ATTRIBUTES:

- string username – user's name
- int age – user's age
- double weight – user's weight
- string fitnessGoal – user's goal (Lose Weight, Build Muscle, Endurance, etc...)
- App myFitness – instance of App for user

### METHODS:

- void display() - displays information about a user
- void saveUserData(string filename) - saves user info and App info to a file
- void loadUserData(string filename) - loads user info and App info from a file
- void startSession() - launches main menu for user

\*constructors, destructors, and getters/setters are **NOT** explicitly listed in **METHODS** but should be implemented and used where needed.

## What the user can do:

---

=====

USER

=====

1. Load User Profile
2. Create New User
3. Exit

=====

MAIN MENU

=====

1. Create Workout
  2. View All Workouts
  3. View by Category
  4. View Favorites
  5. Log Workout
  6. Delete Workout
  7. Save
  8. Load
  9. View User Profile
  10. Return to Main Menu
- 

## ACTIONS BREAKDOWN

1. Create Workout	App::createWorkout()
2. View All Workouts	App::viewAllWorkouts()

\*constructors, destructors, and getters/setters are **NOT** explicitly listed in **METHODS** but should be implemented and used where needed.

3. View by Category	App::viewByCategory()
4. View Favorites	App::viewFavorites()
5. Log Workout	App::logWorkout()
6. Delete Workout	App::deleteWorkout()
7. Save	User::saveUserData()
8. Load	User::loadUserData()
9. View User Profile	User::display()
10. Return to Main Menu	Same as exit

\*constructors, destructors, and getters/setters are **NOT** explicitly listed in **METHODS** but should be implemented and used where needed.