# Homework 05 - Sports!

## Definition/Description

You have been tasked with setting up a system for holding Olympic boxer and swimmer information. Data that you will hold for each of these athletes includes:

> **Boxers**: name, date of birth, country of origin, a list of medals they've won, their weight class, and their record. If a boxer loses 10 fights, they must retire.
> **Swimmers**: name, date of birth, country of origin, a list of medals they've won, and the strokes they know.

It has been decided that all this data will be modeled via a Python package called athlete. Internally, there will be three modules:

> **Athlete.py**, which includes a parent class Athlete
> **Boxer.py**, which includes a class Boxer, inheriting from Athlete
> **Swimmer.py**, which includes a class Swimmer, inheriting from Athlete

It is your job to create this package. (Note: you do not have to hold actual athlete information.)

## Requirements

### Class Athlete

Represents a generic athlete.

#### Class Attributes

1. **athlete_count**
    a. The number of athletes created.

#### Instance Attributes

1. **self.name**
    a. the name of the athlete (String)
2. **self.dob**
    a. the date of birth of the athlete (String)
3. **self.origin**
    a. the country of origin of the athlete (String)
4. **self.medals**
    a. a list of medals the athlete has won (list)

### Instance Methods

1. **__init__()**
    a. Description: Constructs a new Athlete object.
    b. Parameters: 4
        i. name_param (String)
        ii. dob_param (String)
        iii. origin_param (String)
        iv. medals_param (list)
    c. Returns: 0
    d. Assigns parameters to instance attributes. Increases athlete_count by 1.
2. **Getters for all four instance attributes (name, dob, origin, medals)**
    a. Description: retrieves an attribute
    b. Parameters: 0
    c. Returns: 1
        i. self.attribute
3. **add_medal()**
    a. Description: adds a new medal to the medal list
    b. Parameters: 1
        i. medal_param (String)
    c. Returns: 0

## Class Boxer

Represents a boxer athlete.

### Class Attributes

All inherited class attributes, plus:

1. **boxer_count**
    a. The number of boxers created.

### Instance Attributes

All inherited class attributes, plus:

1. **self.weight_class**
    a. the weight class of the boxer (String)
2. **self.record**
    a. the fight record of the boxer (list with two items: [wins (int), losses (int)] )

### Instance Methods

All inherited instance methods, plus:

1. **__init__()**
    a. Description: Constructs a new Boxer object.
    b. Parameters: 5
        i. name_param (String)
        ii. dob_param (String)
        iii. origin_param (String)
        iv. medals_param (list)
        v. weight_class (String)

    c. Returns: 0
    d. Assigns parameters to instance attributes. Increases boxer_count by 1.
2. **__str__()**
    a. Description: retrieves data about the boxer when printing.
    b. Parameters: 0
    c. Returns: 1
        i. Data about the boxer. (String)
3. **Getters for the two new instance attributes (weight_class, record)**
    a. Description: retrieves an attribute
    b. Parameters: 0
    c. Returns: 1
        i. self.attribute
4. **set_weight_class()**
    a. Description: sets an attribute
    b. Parameters: 1
        i. weight_class_param (String)
    c. Returns: 0
5. **win_fight()**
    a. Description: adds one to the wins of the boxer's record
    b. Parameters: 0
    c. Returns: 0
6. **lose_fight()**
    a. Description: adds one to the losses of the boxer's record, then checks to see if the boxer needs to retire (after 10 losses)
    b. Parameters: 0
    c. Returns: 1:
        i. A message about the number of fights left before retirement, or 'This boxer has retired.' (String)

# Class Swimmer

Represents a swimmer athlete.

## Class Attributes

All inherited class attributes, plus:

1. **swimmer_count**
    a. The number of swimmers created.

## Instance Attributes

All inherited class attributes, plus:

1. **self.strokes**
    a. the strokes that the swimmer knows (list)

## Instance Methods

All inherited instance methods, plus:

1. **__init__()**
    a. Description: Constructs a new Swimmer object.
    b. Parameters: 5

        i. name_param (String)
        ii. dob_param (String)
        iii. origin_param (String)
        iv. medals_param (list)
        v. strokes (list)
    c. Returns: 0
    d. Assigns parameters to instance attributes. Increases swimmer_count by 1.
2. **__str__()**
    a. Description: retrieves data about the swimmer when printing.
    b. Parameters: 0
    c. Returns: 1
        i. Data about the swimmer. (String)
3. **get_strokes()**
    a. Description: retrieves the strokes attribute
    b. Parameters: 0
    c. Returns: 1
        i. self.strokes
4. **add_stroke()**
    a. Description: adds a new stroke to the swimmer's repertoire. Checks to make sure the stroke is not already in the list
    b. Parameters: 1
    c. Returns: 0

## Structure

The directory structure should look like the following:

```
ITP_216_H05_LastName_FirstName/
        athlete/
                __init__.py
                Athlete.py
                Boxer.py
                Swimmer.py
        ITP_216_H05_LastName_FirstName.py
```

# Provided Files/Data

None

# Sample Output

**Mary Berry is a Light Flyweight boxer from UK born on 1935/03/24. Mary Berry has a 0-0 record, and has won 2 medals: ['Gold (2012)', 'Gold (2016)'].**

**Dave Thomas is a swimmer from USA born on 1932/07/02. Dave Thomas knows ['freestyle', 'breaststroke'], and has won 2 medals: ['Silver (1992)', 'Gold (1996)'].**

# Deliverables

Compress *your project directory* and submit the resulting zip file with the following name:

**ITP_216_H[number]_LastName_FirstName.zip**

# Grading

| Section | Points (Total: 10) |
|---|---|
| **Functionality and User Interface**<br>1. Imports package and uses it to create one Boxer and one Swimmer. | 0.5 (0.5 points each) |
| **Code**<br>1. Athlete<br>a. __init__()<br>i. correctly set up attributes<br>b. getters (4)<br>i. correctly set up<br>c. add_medal()<br>i. adds new medal to the list<br>2. Boxer<br>a. __init__()<br>i. correctly inherited via super()<br>ii. correctly sets up attributes<br>b. __str__()<br>i. returns String<br>c. getters and setters (3)<br>i. correctly set up<br>d. win_fight() and lose_fight()<br>i. modifies record and returns info when appropriate<br>3. Swimmer<br>a. __init__()<br>i. correctly inherits via super()<br>ii. correctly sets up attributes<br>b. __str__()<br>i. returns String<br>c. get_strokes()<br>i. correctly returns attribute<br>d. add_stroke()<br>i. modifies strokes attribute when appropriate<br>4. __init__.py | 7.0 (0.5 point each) |

| | |
|---|---|
| a. correctly sets up package | |
| **Documentation and Formatting**<br>1. Concise and useful commenting in your codebase is a must. You will need a header with your name, the semester, the section of the course you are in, and the homework number.<br>2. You need descriptions of any major sections in your code (functions, classes, methods, et al.).<br>3. Your code must be generally clear and readable. | 1.5 (0.5 point each) |
| **Error Handling**<br>1. Code interprets without crashing. | 1.0 (1 point each) |