

COMP1406 - Assignment #1

(Due: Thursday, January 24th @ 9:00pm)



In this assignment, you will create 3 objects and get them to interact together. You will add to your code in the next assignment.

(1) The Dive, Performance and Athlete Classes

You will need to define 3 objects as indicated below. Use the diagram at the end of the assignment to determine appropriate attribute names and to ensure that you are storing objects and data in the correct location with respect to the other objects.

- Define a class called **Dive** that will represent a kind of dive in a diving event. Each dive should have a special name (e.g., "reverse pike") and a number (integer from 1 to 4) which represents the difficulty of the dive. Make appropriate constructors.
- Define a class called **Performance** that will represent a particular athlete's diving attempt. Each performance is judged and the judges' scores are recorded. Assume that the diving event has exactly 8 judges. A performance should keep track of a **Dive** object (which is the dive being performed) as well as an array of exactly 8 judges' scores (which are each floats in the range of 1.0 to 10.0). Make appropriate constructors.
- Define a class called **Athlete** that will represent a particular athlete who may perform in various diving events. Each athlete should have a name, a country which they are representing and a list of the diving performances that they gave (an array that can hold exactly 3 **Performance** objects, initially empty). You will also want to keep track of the latest performance number (to be used as an index to the array). Make appropriate constructors. Create a method called **perform(Dive)** that takes a **Dive** object as a parameter and simulates the athlete actually performing the dive. This method should create a new **Performance** object for the athlete's performance array. If this method is called after the athlete has already performed three times, it should simply do nothing. Create a method called **printResults()** that displays the results of the athlete's diving performances like this:



```
Bob's diving performances:  
0 - reverse pike  
1 - reverse triple twist  
2 - double forward with triple somersault
```

- Add the following **static** constant arrays to the top of your **Dive** class:

```
static final String[] diveNames = {"reverse pike", "forward pike",  
    "reverse armstand with double somersault", "reverse triple twist",  
    "double forward with triple somersault", "cannon ball"};  
  
static final int[] diveDifficulties = {3, 2, 2, 4, 4, 1};
```

- Create a **static** method in the **Dive** class called **chosenRandomly()** that returns a new **Dive** object with a name and difficulty chosen randomly from the **diveNames** and **diveDifficulties** arrays (just created above). Note that the difficulties above "correspond to" (or match up with) the names. That is, a reverse pike is always difficulty 3 and a cannon ball is always difficulty 1.

(2) Understanding How Objects Are Stored

Write a program called **DiveTestProgram1** that constructs the object structure shown in the diagram on the next page, storing the three constructed athletes into variables called **art**, **dan** and **jen**. Add to the end of the program code that does the following:

- Prints the performance results of art, then dan, then jen
- Starting with the variable art, write code that accesses and prints out the difficulty of the 2nd dive that was performed (should be 1).
- Starting with the variable dan, write code that determines and prints out the number of dives that dan performed (should be 2). You will need to go through the array and count the dives.
- Starting with the variable jen, write code that accesses and prints out the 6th score of her first dive (should be 0).
- Use the **chosenRandomly()** method to create a dive and store it in a variable. Then access and print the stored Dive's name and difficulty. Note that the output should differ each time that you run the program.

Here is an example of what you should see as output:

```
Art Class diving performances:
```

```
0 - reverse pike
1 - cannon ball
2 - reverse triple twist
```

```
Dan Druff diving performances:
```

```
0 - reverse triple twist
1 - forward pike
```

```
Jen Tull diving performances:
```

```
0 - forward pike
```

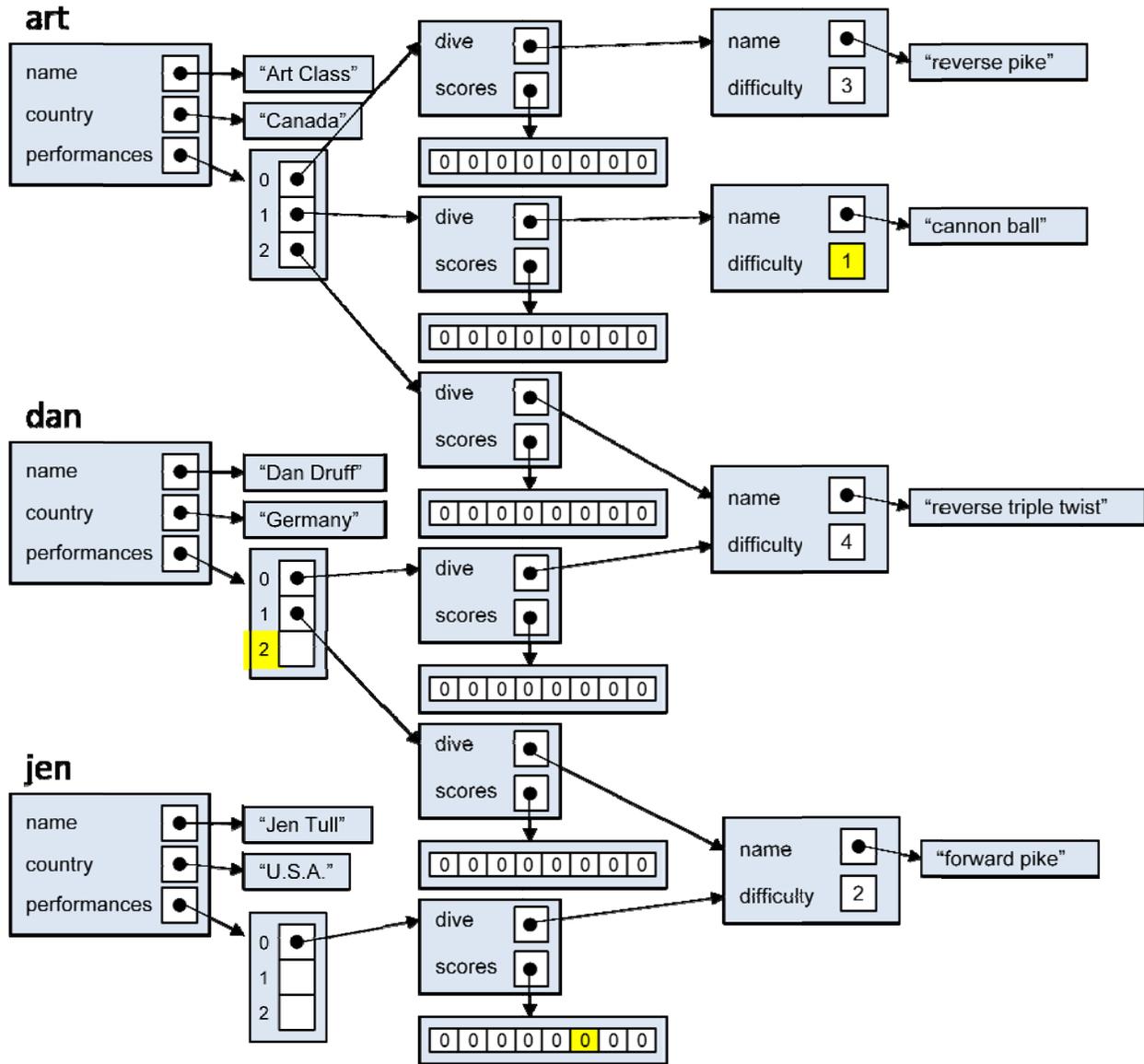
```
Art's 2nd dive difficulty: 1
```

```
Dan's number of dives: 2
```

```
Jen's 6th score of dive 1: 0.0
```

```
Random dive's name:          double forward with triple somersault
```

```
Random dive's difficulty: 4
```



NOTE: Submit all **.java** and **.class** files needed to run. You **MUST NOT use packages** in your code, **nor projects**. Submit ALL of your files in one folder such that they can be opened and compiled individually in JCreator. Some IDEs may create packages and/or projects automatically. You **MUST** export the **.java** files and remove the package code at the top if it is there. Do NOT submit JCreator projects either. **JUST SUBMIT the JAVA and CLASS FILES.** Note that if your internet connection at home is down or does not work, we will not accept this as a reason for handing in an assignment late ... so make sure to submit the assignment WELL BEFORE it is due !

Please NOTE that you WILL lose marks on this assignment if any of your files are missing. You will also lose marks if your code is not written neatly with proper indentation. See examples in the notes for proper style.
