

## Spring 2026 COP 3330 Program #4: Arrays, Objects

### Class A: Figure Skating Scoring (figureskating.java)

This portion of the assignment won't be a full program. Instead, you will fill in methods in a class. You will be responsible for creating some tests. We will grade the correctness of your methods by calling them from a different class, so it's critical that you call your class figureskating and that you use the exact method names provided below.

#### Method #1: judgeScore

A score for a figure skating routine is based on the sequence of maneuvers the participant attempts in the routine. Each maneuver has a difficult level (real number) and each judge will assign a score (real number) for each maneuver. The total score assigned by that judge will be the sum of the products formed by each difficulty and corresponding score. For example, if a participant tries a sequence of 4 maneuvers with difficulties 2, 3, 1 and 6, and a judge gives that participant the scores of 4, 3, 2, and 5 for those maneuvers, respectively, that judge's score for the participant would be  $2 \times 4 + 3 \times 3 + 1 \times 2 + 6 \times 5 = 8 + 9 + 2 + 30 = 49$ . You will write a static method that performs this calculation, given both the difficulty ratings of the maneuvers and the scores given for those maneuvers by a judge, which returns that judge's score for the routine.

#### Method #2: sanitizeScores

It's typically the case that in some figure skating competitions, many judges give scores to a participant, but usually some number of the lowest scores and some number of the top scores are dropped. For example, if the scores a participant received were 10, 8, 6, 11, 9, 10, and 12 and we were dropping the 2 lowest scores and the 1 highest score, the valid list of scores (in sorted order) would be 9, 10, 10, and 11. (The low scores of 6 and 8 were dropped, as well as the high score of 12.) You will write a static method that takes in an array of all of the original scores, the number of low scores to be dropped and the number of high scores to be dropped and returns a new array storing the scores that count (the ones except the dropped scores) in sorted order.

Here is a framework to fill out:

```
import java.util.*;
public class figureskating {

    // Returns a judge's score for a routine where difficulties stores the
    // difficulties of each maneuver and scores stores the corresponding scores
    // the judge gave for each maneuver. Both arrays are guaranteed
    // to be the same size.
    public static double judgeScore(double[] difficulties, double[] scores) {
        // Fill in code.
    }

    // Returns the remaining scores in sorted order after dropping numLowDrop
    // lowest scores and numHighDrop highest scores from allScores. No change
    // should be made to the allScores array.
    public static double[] sanitizeScores(double[] allScores, int numLowDrop,
                                         int numHighDrop) {
        // Fill in code.
    }
}
```

### **Program B: Using the Time\_V1 class**

Write a program that uses the Time\_V1 class shown in class. Think of an application program for which Time objects might be useful. If you want to use more than one class other than the Time\_V1 class, feel free to do so. The key here is to think of something you're interested in and create one or more Time\_V1 objects in your main code, then call methods from the Time\_V1 class as necessary. To illustrate limitations in real life code, **you are not permitted to modify the Time\_V1 class.**

### **Implementation Requirements**

For Program A, please add a main method in our figureskating.java class where you show your tests of the two methods shown. A portion of your grade will be based on how thorough the TAs deem your tests to be. Make sure to all the methods the correct names and the class the correct name since the TAs will be calling these methods from a different class.

For Program B, **do NOT submit the Time\_V1 class.** The TAs will use the version of the class posted online. Instead, just turn in the other class (or classes) for your program. In the header comment of your file(s) indicate which file has the main method that should be run. For this portion of the assignment, you may call your files and classes whatever you want, but do try to make meaningful names.

### **Deliverables**

For part A, **please submit a single .java file called, figureskating.java** with the two required static methods and a main method with your tests.

For part B, submit one or more .java files for your program that uses at least one Time\_V1 object.

Make sure to include a header comment for each file you submit and appropriate internal comments.