

---

## COS 485: Program #8 – Team Balancer

**Objectives:** Designing an iterative improvement algorithm (suggestion, but do what you want)

---

Abu 97	Ace 70	Ada 35	Ali 80	Amy 76	Ana 33	Ann 68	Ari 89	Asa 43	Ava 75
Avi 50	Axl 50	Aya 61	Ben 59	Bev 40	Bob 65	Cam 77	Dan 86	Dax 32	Deb 62
Dot 50	Ely 59	Eva 28	Eve 51	Fin 53	Flo 44	Gia 80	Gus 73	Hal 51	Ian 70
Ida 54	Ima 37	Ira 37	Ivy 74	Jad 29	Jan 59	Jax 58	Jay 65	Jim 55	Joe 82
Jon 68	Joy 68	Kai 37	Ken 70	Kim 78	Kit 63	Kya 62	Kye 71	Lea 72	Lee 53
Leo 73	Lia 40	Lil 54	Liv 51	Luz 48	Mae 49	Max 52	Meg 94	Mel 53	Mia 69
Moe 59	Mya 37	Neo 88	Nia 51	Noa 74	Noe 62	Obi 71	Pam 67	Rae 52	Rex 74
Roy 78	Sam 34	Sky 76	Tia 71	Van 52	Vic 51	Wei 69	Xao 41	Zak 93	Zoe 83

Design an algorithm to assign players to sports teams so that every team gets the same number of players, and also so that the sum of skill levels on each team are as evenly balanced as possible.

In the example above there are 80 players, and we want 8 teams of 10 players each. The names and skill levels are shown. The players are arranged in 8 rows of 10 players, but if these were the teams, the skill level sums would not be well balanced. You can do a lot better!

Don't expect to find an optimal solution but try to find something good.

The input is:

- an array of Player objects (with fields: name and skill)
- the number of teams to create

You may assume that the number of players will be evenly divisible by the number of teams, and that the skill levels are integers in the range from 0 to 100.

### Setting up the project in Eclipse:

Create a new project similar to how you set up program 1:

- It will use TeamBalancer.jar, the same Scaffold jar, and starting code TeamBalancer.java
- In the run configuration set **Main class** to: tester/TeamBalancerTester

### What to turn in:

#### Written Report turned in through Brightspace

1. A screen shot of the result tab for test80.txt
2. A short well written English explanation of your algorithm.

### Electronic Submit

From a Unix machine in the lab run the program “submit” to submit your files.

Submit your source code (.java files) and compiled code (.class files) to the directory: **prog7**

### Grading:

- 80 points – Finding a good solution
- 20 points – Your algorithm explanation
- 10 points extra credit – for finding the best solution in the class