

ACS-2947-050

Lab #6

Due by Friday November 9 at 11:59 pm

Submit your .java files to [2947L-070@acs.uwinnipeg.ca](mailto:2947L-070@acs.uwinnipeg.ca) or [2947L-071@acs.uwinnipeg.ca](mailto:2947L-071@acs.uwinnipeg.ca)

Comparable/Comparator [1904 notes](#) (Interfaces)

Also see p. 103-118 [1904 text](#)

In the regular season of the NHL, games are played in 60 minutes. After the regulation time has elapsed, the team with the most goals scored wins. If the score is tied at the end of the game, there is an additional 5-minute overtime period where the team who scores the next point wins. If no overtime goal is scored, the winner is determined in a shootout.

In the overall standings, teams earn 2 points for a win, 1 point for a loss in overtime/shootout, and 0 points for a loss in regulation time i.e. teams earn at least 1 point if tied at the end of the game.

Refer to the following sample data:

Team		W	L	OTL	ROW
Winnipeg	Jets	52	20	10	48
Chicago	Blackhawks	33	39	10	32
Dallas	Stars	42	32	8	38
Nashville	Predators	53	18	11	42
Colorado	Avalanche	46	27	9	40
Minnesota	Wild	45	26	11	41
St. Louis	Blues	44	32	6	41

1. Create a `Team` class which includes:
  - a. instance variables that store the following information:
    - A team's name: `String city, String name`
    - number of wins: `int wins`
    - number of losses: `int losses`
    - number of overtime/shootout wins: `int otl`
    - number of regulation overtime wins: `int row`

- b. methods that calculate and return the following:
    - `gamesplayed()` that returns an `int` of the total number of games played (each win (W), loss (L), and overtime/shootout loss (OTL) represents a game played)
    - `points()` that returns an `int` of the total points for each team: a team earns 2 points for each win, 1 point for each overtime/shootout loss, and 0 points for each loss
2. Implement your own concrete Comparator class for comparing `Team` objects called `TeamComparator`.
  - You will need to implement the `java.util.Comparator` interface
  - The comparison rule should be as follows:
    - The team with the higher total points is greater than the other team.
    - In the event of a tie, then the comparator will look at the number of games played. The one with fewer games played is considered the greater team.
    - If they have the same number of games played then the regulation overtime wins (ROW) is considered. The team with higher ROW is considered greater.
3. Fill an `ArrayList` (`java.util.ArrayList`) with `Team` objects with data from the sample above.
4. Using the `ArrayList` of `Team` objects:
  - a. ***Iterate through*** the list and find the best `Team` using your `TeamComparator`. Display the current leader in the division.
  - b. Sort the list using `Collections.Sort` and display the top 3 teams in the division.