# Data Structures and Java (Spring 2015)

## Assignment 1 – Arrays, Strings and Loops

### Assignment objectives:

In this assignment, you may use your knowledge of the following concepts:

1. Arrays – accessing elements in a location, finding the length of the array
2. String – length(), toCharArray(), charAt(), substring()
3. Making a simple method call and sending an input (known as argument)
4. For loop, while loop, if-else
5. Math (+, %, ++, +=) and relational operators (!=, ==, <,>)

In the template provided (located in the git repository under “assignments/A0”, the code requests a play by play report for a single game from the NHL.com. The response contains the HTML that will be used for this assignment. The report used is for a game between the Tampa Bay Lightning and the Chicago Blackhawks and is located here: <http://www.nhl.com/scores/htmlreports/20132014/PL020026.HTM>

In the play by play report there are details for every event in the game that the NHL tracks. In this assignment, we’ll be working with face-off events. To get a sense for what a face-off is, have a look at the first 10 seconds of this video of a game between the San Jose Sharks and the Vancouver Canucks: <http://www.youtube.com/watch?v=LgAGMCvPE1E>

### Tasks

We’ll be working with the 3rd face-off of the game. Specifically, all of your output should reflect the 3rd face-off only. **Note, however, that your logic needs work for the 5th (i.e. nth) faceoff of the game which involves different players. We will test your output with the 3rd faceoff and the 5th faceoff. You should do the same.**

The HTML for the 3rd face-off is as follows:

<td class=" + bborder">CHI won Neu. Zone - T.B #91 STAMKOS vs CHI #26 HANDZUS</td>

The HTML for the 5th face-off is as follows:

<td class=" + bborder">T.B won Off. Zone - T.B #9 JOHNSON vs CHI #16 KRUGER</td>

#### Task 1

Create a new project in NetBeans with a class called A1FirstNameLastName.java. Integrate the code provided in the template.

#### Task 2

Remove the HTML tags from the eventText variable and store it in a new variable called eventTextNoHTML.

**Output:** The value of eventTextNoHTML is: \_\_\_\_\_\_\_\_\_\_\_

#### Task 3

Take the variable eventTextNoHTML and put it into a character array (char[] eventTextNoHTMLCharArr).

**Output:** The length of the array eventTextNoHTMLCharArr is: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Output:** The second character in eventTextNoHTMLCharArr is: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#### Task 4

Using a for loop, examining one character at a time, count how many special characters (i.e. non-letter, non-digit) there are in eventTextNoHTMLCharArr.

**Output:** The number of special characters in eventTextNoHTMLCharArr is: \_\_\_\_\_\_\_\_\_\_\_

#### Task 5

Using a while loop, examining one character at a time, count how many spaces there are between the beginning of the String variable eventTextNoHTML and the first player’s last name in the faceoff, i.e. STAMKOS in the case of the 3rd face-off. Hint: your loop breaking condition should be the first space you find after the first “#” character you examine.

**Output:** The number of spaces until we reach the first player’s last name is: \_\_\_\_\_\_\_\_\_\_\_

#### Task 6

Extract the following data from eventTextNoHTML and store them in separate variables:

**CHI won Neu. Zone - T.B #91 STAMKOS vs CHI #26 HANDZUS**

* Which team won the faceoff, e.g. CHI (String winningTeam)

**HINT: winning team is always the first 3 characters**

* The zone of the faceoff, e.g. Neu. Zone (String faceoffZone)

**HINT: face off string always starts at position 8 and goes for 9 characters (i.e. Neu. Zone, Off. Zone, Def. Zone)**

* The first player’s number, e.g. 91 (int firstPlayerNumber)

**HINT: Search for # and then for the first space after #**

* The first player’s team, e.g. T.B (String firstPlayerTeam)
* **HINT: Get the first “- “and space before first #**
* The first player’s last name, e.g. STAMKOS (String firstPlayerLastName)

**HINT: Get the first and second space after firsthash.**

* The second player’s team, e.g. CHI (String secondPlayerTeam)

**HINT : Get the data between “vs “ and the space after that**

* The second player’s number, e.g. 26 (int secondPlayerNumber)

**HINT: Get the “#“ after the first #**

* The second player’s last name, e.g. HANDZUS (String secondPlayerLastName)

**Output (for each variable above):** The value of <variable name> is: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Output:** The first player’s number is [greater than | less than | equal to] the second player’s number.

**Output (for the second player only):** The length of the string in the variable secondPlayerLastName is: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#### Task 7

Take the first player’s last name and format it with an uppercase first letter followed by lower case letters. For example, STAMKOS becomes Stamkos. Store this in a variable called firstPlayerLastNameFormatted. Do not worry about logic for players with multiple words in their last name, e.g. VAN REIMSDYK would simply become Van reimsdyk.

**Output:** The value of firstPlayerLastNameFormatted is: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#### Task 8

Use a StringBuilder to put output the following .csv row from the real values that you’ve extracted:

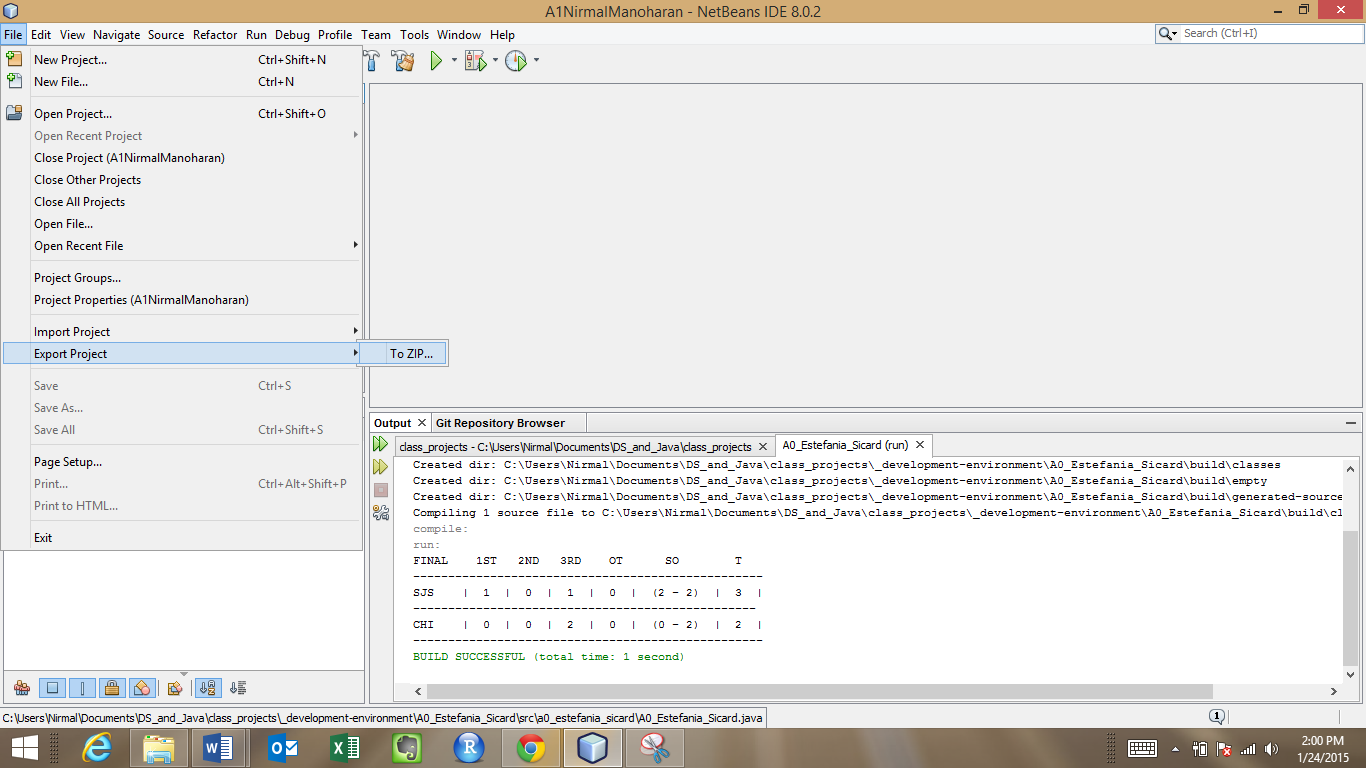
winningTeam, faceOffZone, firstPlayerNumber, firstPlayerTeam, firstPlayerLastName, secondPlayerTeam, secondPlayerNumber, secondPlayerLastName

**Output:** The .csv row is**: CHI,Neu. Zone,91,T.B,STAMKOS,CHI,26,HANDZUS**

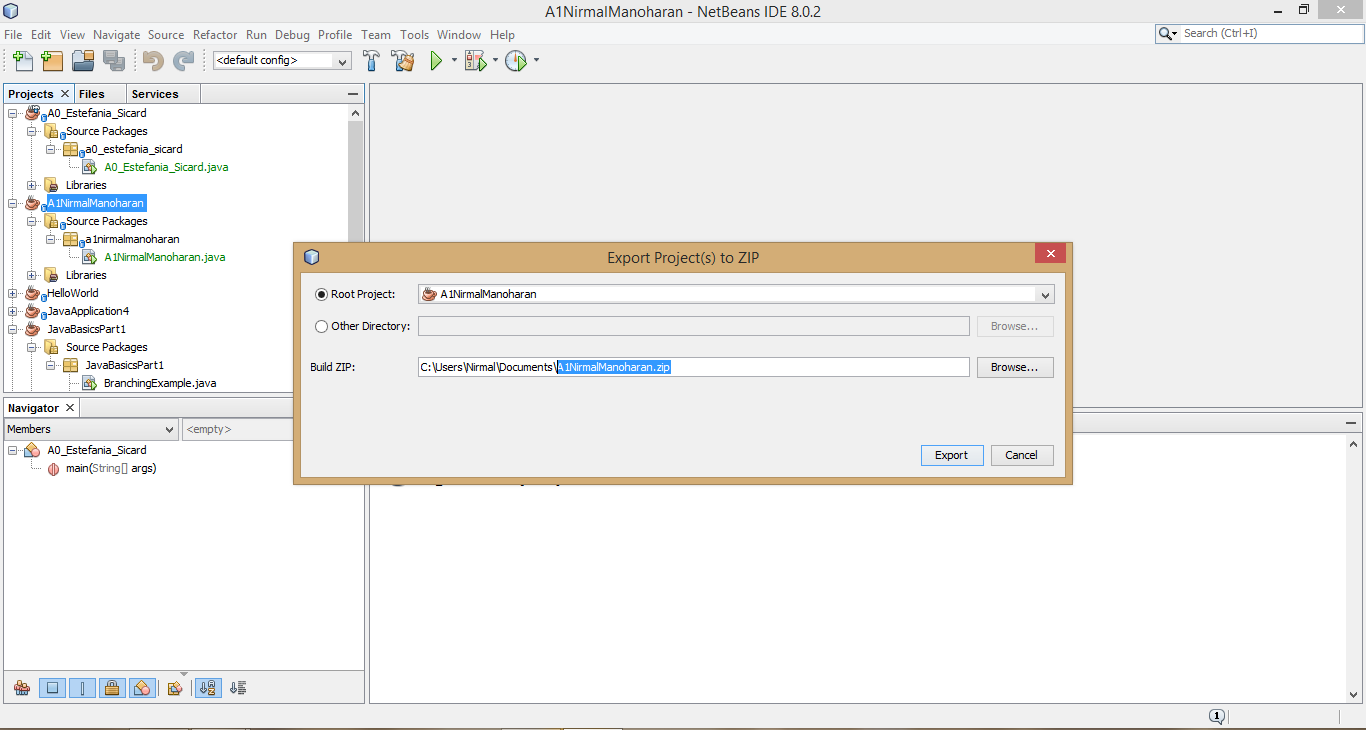
### Deliverables

Please export your project using Export option in File Menu. (Screenshots below).

1. Click File > Export Project > To ZIP..



1. Select the project to export under Root Project. Specify the path and ZIP file name in Build ZIP field. Name the zip file **A1FirstNameLastName.zip** and click Export.



1. Upload ZIP file to Canvas.