Team Practice 1

Getting Started

This assignment makes use of the class String as well as two if statements (conditionals). Work out the logic involving only strings first, and then complete the code that requires if statements.

Employee Info

You will be writing a program that reads a line of text from the user containing information on an employee and outputs specific information from that input in a specified format. The following information, **in the order shown below**, will be included in the input text. Note there can be additional text before/after each element (see examples on the next page). You may assume **there is always one and only one space between words in the input**.

1. The input will begin with a sequence of words and spaces.
2. The employee's first name and last name will always appear after a colon (the only one in the input) and there will always be a comma after the last name.
3. The employee's id number, which will be preceded by the pound sign (#) and followed by a comma. This will be the only pound sign in the input.
4. The number of dollars paid to the employee, which will be preceded by the word "paid" and followed by the word "dollars." These will be the only occurrences of those two words in the input.
5. The number of cents paid to the employee, which will be preceded by the last comma in the input and followed by the word "cents." The value of cents will be either one or two digits.
6. The phrase "per hour," which will only appear in the input if the employee is paid by the hour. This phrase may be in uppercase, lowercase, or mixed case.

The output from your program will be:

- The employee id.
- The name of the employee in uppercase
- The employee's pay as dollars and cents, with two digits after the decimal point.

The employee's status, which is "hourly" if the employee is paid by the hour or "salaried" otherwise.

Sample interactions below demonstrate how the program should run, with user input shown in **boldface**. These samples do not cover all test cases; be sure to test with other inputs as well.

Sample interaction 1:

```
Please enter information on the employee:
Employee info: Stuart Randolph, emp #244891, is paid 15 dollars, 14 cents per
hour worked

ID: 244891
Name: STUART RANDOLPH
Pay: $15.14
Status: hourly
```

Sample interaction 2:

```
Please enter information on the employee:
Info on employee: James taylor, employee #4242424242, is paid 150,000 dollars,
0 cents

ID: 4242424242
Name: JAMES TAYLOR
Pay: $150,000.00
Status: salaried
```

Sample 3:

```
Please enter information on the employee:
Employee: janet jones, #6565657, paid 25 dollars, 5 cents total Per Hour

ID: 6565657
Name: JANET JONES
Pay: $25.05
Status: hourly
```

Hints:

- Copying and pasting an input string from the assignment write-up can be risky, as it could include invisible characters, such as linefeeds. A less error-prone way to avoid having to enter a long text sequence every time you test your program is to hardcode the input by assigning it to a variable in your code. Once it's working with that text, change it and test again. Then update your code to read in text from the user and continue to test.
- Some of the string methods that will be helpful in this assignment include **substring()**, **indexOf()**, **lastIndexOf()**, and **length()**.
- Steps 5 and 6 each require the use of an if statement. For step 5, get your code to work correctly for extracting the amount of dollars and cents before correcting for a cents value that contains just one digit.

  In step 6, there is no indexOf() method that ignores case. The variable `position` will have a value of -1 in the following case: `int position = "Good morning".indexOf("GOOD");`

  A common approach used in matching text that may be written in different cases is to convert the text and the phrase to the same case before looking for the phrase within the text.

- Be sure to test your code at each step of the way, as it is very easy to be off by a character or two. You may find it helpful to print the starting and ending positions of the text to be extracted along with that text and/or use the debugger to check values. You can always remove or comment out any extraneous output once everything is working.