# CSE 2221 - Project 3

Task

Gain familiarity of **while** loops, **for** loops, and **static** methods by computing the de Jager formula on user-inputted values. Will also gain familiarity of nested **for** and **while** loops.

Original Project Instructions

Project 3 Instructions from CSE2221 Project Site

Program Requirements

- Ask the user to input a value for $\mu$
- Ask the user to input a value for $w$
- Ask the user to input a value for $x$
- Ask the user to input a value for $y$
- Ask the user to input a value for $z$
- Approximate the user-inputted value $\mu$ (within a relative error of 1%) using the de Jager formula and the $w$, $x$, $y$, and $z$ user-inputted values
- After approximating $\mu$, display the values of $a$, $b$, $c$, and $d$ resulting from the de Jager formula calculation, the approximation of $\mu$, and the relative error of the approximated $\mu$ versus the actual $\mu$

Summary of de Jager Formula

$\mu \approx w^a x^b y^c z^d$ where $a$, $b$, $c$, and $d$ are values in the array
$[\ -5, -4, -3, -2, -1, -1/2, -1/3, -1/4, 0, 1/4, 1/3, 1/2, 1, 2, 3, 4, 5\ ]$

de Jager Formula Example

$\mu = 238900$
$w = 14$
$x = 102329$
$y = 1936$
$z = 13$

After running the de Jager computation, we get the following values:

$a = -5$
$b = 1$
$c = 1/2$
$d = 4$
$14^{-5} * 102329^{1} * 1936^{1/2} * 13^{4} = 239,103$ which is within about 0.08% of $\mu$

Steps

1. Copy and paste *ProjectTemplate* to create a new project folder for this project
2. Name the project *Pseudoscience*
3. Open the *src* folder, then open *(default package)*
4. Rename *ProgramWithIO.java* to *ABCDGuesser1.java*
5. Delete the other files
6. Open *ABCDGuesser1.java*
7. Update the JavaDoc comments above the class declaration (i.e. program description and author name)
8. Copy and paste the following private static methods above the `main` method

```
/**
 * Repeatedly asks the user for a positive real number until the user enters
 * one. Returns the positive real number.
 *
 * @param in
 *           the input stream
 * @param out
 *           the output stream
 * @return a positive real number entered by the user
 */
private static double getPositiveDouble(SimpleReader in, SimpleWriter out) {

}

/**
 * Repeatedly asks the user for a positive real number not equal to 1.0
 * until the user enters one. Returns the positive real number.
 *
 * @param in
 *           the input stream
 * @param out
 *           the output stream
 * @return a positive real number not equal to 1.0 entered by the user
 */
private static double getPositiveDoubleNotOne(SimpleReader in, SimpleWriter out) {

}
```

9. Complete the above two methods according to their JavaDoc

   Note: You cannot assume the user will enter in a number. Your methods should read the input as a String, check that the user input is a real number, then if the check passes, convert the String user input to a double. The following functions will be helpful:

   - String stringValue = in.nextLine()
   - FormatChecker.canParseDouble(stringValue)
   - double doubleValue = Double.parseDouble(stringValue)

10. Add the necessary code to satisfy the project requirements (i.e. $a$, $b$, $c$, and $d$ using the de Jager formula). I suggest making another private static method for doing this, especially since you will do it in part 2 anyways

    Note: When creating the array for possible values of $a$, $b$, $c$, and $d$, remember $1/4 = 0$ because of integer division! there is a simple fix you need to make to ensure you are doing float / double division

11. Copy and paste *ABCDGuesser1.java* to create a new file, name the new file *ABCDGuesser2.java*

12. Open *ABCDGuesser2.java*

13. Edit *ABCDGuesser2.java* so that your de Jager computation uses **for** loops instead of **while** loops.

    This does NOT apply for the loops in your `getPositiveDouble` and `getPositiveDoubleNotOne` methods! Leave those as **while** loops

14. Also edit *ABCDGuesser2.java* so that it uses an additional private static method

If you did your de Jager formula computation in a separate private static method (like I mentioned in step 10), then this item does not apply to you, you have already satisfied this requirement

15. Create a zip file of your *Pseudoscience* project

16. Rename the zip file (not your project folder) using the naming scheme ”FirstName_LastName_DotNumber_ProjectNumber.zip”, for example mine would be ”Logan_Frank_580_3.zip”

17. Submit to Carmen