

## Strings and Conditionals

### **What To Do:**

Follow each step carefully. As you complete the lab, submit the source files (.java) problems to the autograder (link is in the Canvas portal). After finishing, please submit your work, as a ZIP, to Canvas, and let one of the AIs know.

**For this lab, please place each method inside its own class file labeled as ProblemX, where X is the problem number. The accompanying test files should be named ProblemXTest.**

**Problem 1:**

Design the `String userId(String f, String l, int y)` method that computes a user ID based on three given values: a first name, a last name, and a birth year. A user ID is calculated by taking the the first five letters of their last name, the first letter of their first name, and the last two digits of their birth year, and combining the result. Your method should, therefore, receive two `String` parameters and an `int`. Do not convert the year to a `String`. Below are some test cases.

```
userId("Joshua", "Crotts", 1999)    => "CrottJ99"  
userId("Katherine", "Johnson", 1918) => "JohnsK18"  
userId("Fred", "Fu", 1957)          => "FuF57"
```

**Problem 2:**

Design the `int max(int x, int y, int z)` that returns the maximum of three integers  $x$ ,  $y$ , and  $z$ . Do not use any built-in (Math library) methods.

### **Problem 3:**

Design the `String popChars(String s, char c, char d)` method, which receives a string *s* and two characters *c*, *d*. The method removes *c* if *s* starts with *c*, and removes *d* if the second character of *s* is *d*. The remainder of the string is the same.

### Problem 4:

Design the `String middleString(String a, String b, String c)` method, which receives three strings *a*, *b*, and *c*, and returns the string that is “in between” the others in terms of their lexicographical content. You cannot sort the strings or use an array. Hint: use `compareTo`.