

## Lab 4

Get checked off for lab 3 within the first 10 minutes of lab. You cannot make up any points for lab 3 after this lab.

Next, TAs will distribute a 10-minute quiz for 3 pts followed by a 30-40 minute worksheet for 2 pts. The worksheet will contain core concepts covered in this week's lab, lecture, and assignment, and the TA will go over the worksheet with the class, after you are given some time to work on the sheet in groups. After the TA goes over the worksheet, you will have a 50-60 minute hands-on coding activity for 5 pts that you can complete on your own or in pairs.

**Pair Programming:** In this lab, you can choose a partner for pair programming on the coding portion of the lab. **You must be checked off together, and you only need one computer for pair programming.** One person will be the driver, who controls the computer and codes, while the other is the navigator, who observes and advises. After 20 minutes, you will switch driver and navigator, continuing this pattern until the task is complete. Please read more about pair programming and the benefits: [Pair Programming Student Handout](#)

**In order to get credit for the lab, you need to submit your quiz to a TA and get the worksheet and code checked off by the end of lab. For non-zero labs, you can earn a maximum of 3 points for lab work completed outside of lab time, but you must finish the lab before the next lab. For an extenuating circumstance, contact your lab TAs and Jennifer Parham-Mocello.**

### Problem statement

The Super Bowl is around the corner. As you may notice, the Super Bowl uses roman numerals instead of numbers for the game (see picture below). For example, this year it's Super Bowl LIII, where "LIII" stands for 53 in roman numerals. You are tasked with creating a program that will convert a roman numeral into a decimal number. The program will ask the user for a roman numeral, which is a string, and output its corresponding value in decimal number. No error handling is needed and suppose all the user inputs are valid. However, the user should be able to convert as many roman numerals as he or she wants in a single run. In other words, the program should not end unless the user tells you to.



Roman Numeral conversion rules and chart can be found [here](#):

\*\*\*Note: Ignore the bar rule (multiply by 1000).

## **Worksheet (2pts)**

First, get into groups and spend 10 minutes thinking of the design to the problem. Then, spend 5 minutes implementing your design.

After that, your TA will explain his or her algorithm to the problem as well as demoing the code.

Now, your TA will show you how to solve this problem by using functions.

**Answer the following questions on a piece of paper with your name written on the paper to give your TA:**

1. Why functions are useful and what is refactoring?
2. In your own words, explain arguments, parameters, preconditions, and return values.

## **It's Your Turn: More loops and start using function**

### **Problem Statement**

Create a program that will convert a decimal value into roman numeral, and output the result. The program will ask for an integer input, assume all inputs are valid, and the program should not end unless the user tells you to.

### **(2 pts) Design**

Write out your design to the problem statement above. You will implement it later.

### **(2 pts) Implementation**

**If you are in lab, get your design checked off by a TA before proceeding to code**

Write a program called **decimal\_to\_roman.cpp** to implement your solution to the problem stated above. You may write all of your code in main function at this moment.

### **(1 pt) Refactoring**

Once your program is running correctly, put your conversion algorithm into a function named **d\_to\_r()**, which takes an int as parameter and return a string. You are allowed to create more helper functions if needed.

**Show your completed work to the TAs for credit. You will not get points if you do not get checked off!**