(0) Download the Lab8 Starter Kit from onCourse. SAVE the .zip on you Desktop and unzip the file.
(1) Create a new (console) project.
(2) Learn to generate pseudo-random numbers. Google `srand()` and `rand()`.
   (a) Generate a random integer between 1 and 4, inclusive
   (b) Put (a) in a loop to generate ten (10) random numbers in this range; print two headings separated by commas and then successive rows of output of a counter of the i^{th} iteration you are on and the new random number with the two values separated by commas, e.g.

```
iteration, random#
1, 4
2, 1
3, 2
  :
```

   (c) Add a third column to your .csv output. Within your while loop, write a switch() statement to check the random number that was generated and then act on each case (1, 2, 3, 4, or default). (*Note: we really never should get any other value other than 1 to 4, but I strongly recommend that you always write a* default *case in your switch statement … just in case! If your switch does enter the* default *case, just print a message like:* "Huh? Why am I in default of switch? The value is ____."

Each case of your switch should print the English word that matches the random number generated, e.g.,

```
iteration,  random#,  word
1,  4,   four
2,  1,   one
3,  2,   two
      :
```

(3) Write your results to a `.csv` output file. Open a comma-separated value file called "`output.csv`" (note the `.csv` file extension) and write your output to this file rather than to the output console. Once your program finishes, double-click the file. It should open in Excel, right? (*Note: I've given you a sample C++ file as a reminder and reference for opening files in C++; this is a generic example, not exactly what you need for this lab*).
(4) In Excel, save a new version of your .csv output file. Make sure you SAVE AS (an Excel worksheet, .xls extension). If you make fancy formatting in a .csv file, you'll lose your formatting ☹.
(5) Using Excel, make a graph showing a line of your random numbers generated (yeah, your graph should bounce around between 1 and 4, right? Copy your graph to Word. Add a Figure legend.

Work on your program a3

(6) Get your JOB.h and JOB.cpp code started. Write a printJOB() method to print the contents of a job object. Can you get a main() working to declare a sample JOB and print out its contents?

```
main()
{
     JOB one("print",  512,  5,  5);
     one.printJOB();
}
```

(7) Read your a3.spec. Work on Part I where you generate five(5) JOBs and put them on an STL queue of JOBs. On to Part II and III …