

Week 1 Lab: Introduction

As the first lab of the course, this one is a little different from the others because you have not yet started on a backlog. In this lab you should make sure you are ready to work on the lab. To do so complete the following:

1. Make sure that you have either Ubuntu or Cygwin on your laptop.
2. Download the arrays backlog from ds.nathanielgmartin.com.
3. Download the initial files for the backlog from ds.nathanielgmartin.com using “`wget ds.nathanielgmartin.com/wk1-arrays.zip`” from your Unix environment.
4. Unzip the files in your Unix environment using “`unzip wk1-arrays.zip`”.
5. Start work on the backlog for Section A.

The sprint backlog for next week are stories 1, 2, and 3.

Subsequent labs will be demos as they were in Structured Programming. That is, you will demonstrate the programs you have written over the course of the week and explain how those programs work to your teacher.

Please do not copy other student's files

You may get help from others in your class, but if you simply copy their code, it will be obvious when you try to explain how it works in the demo. Duplicate answers may not be given credit, so you may lose credit if you give your answers to another student, even if you did the work.

We encourage you to talk to each other about your programs—you can learn a lot from your fellow students—but you learn nothing by turning in another students work. So discuss the questions and discuss your answers, but do not copy answers.

Practical Grade

At the end of the semester, the lab class instructors will calculate the practical marks for the class by calculating the percentage of the total marks for all the demos and multiplied by the maximum practical marks for the class.

Tips

1. **Use Test Driven Development.** You will receive a zipped file that contains a lot of code. The first thing you should do is compile it and run the tests against it. If the tests work, read the code that implements the story and make sure you understand how it works. Once you understand how the code work enough to re-implement it yourself, you have completed the story.

2. **Code must be formatted correctly.** You get marks for correctly formatting your code. The code you download is formatted correctly.
3. **Code must compile without warnings.** You will receive marks for code that compiles. The code you download compiles without warning.
4. **So, start early and build good habits.** Easy marks are available now; later they will be harder to earn.
5. **Add 1 or two lines and make sure the formatting and compilation are correct before moving adding more lines.** In this way, you can always be sure that you can get maximum marks for formatting and compilation. It will also make it easier to program.
6. **Do the stories in order and test as you go.** The stories are arranged so you can use the earlier stories to test them. If you make sure that the early stories work, it will be easier to complete the later stories.