

# Lab 3 blank

Instructions:

The following lab is due **10 minutes *before* the start of your next class with me.**  
**Please use your given name + preferred name if they are different:**

Your full name : \_\_\_\_\_

Your Student Number: \_\_\_\_\_

## Submission Process:

Notion is a tool which allows you to easily paste code and submit it to me. Go to <https://www.notion.so/> to make an account, and then once logged into notion, create a page which you will use to answer the questions in this challenge.

You will see an area in the top right corner that says "Duplicate". Click this.



You will now have a copy of the Lab instructions within your Notion account. Fill in the answers to the questions.

Click the **Publish to web** button.



## Publish to web

Publish a static website of this page. You can allow others to view, duplicate, and remix.

Publish to web

Next, turn on **Allow editing & Allow comments**. Then click **Copy web link**:

Share **Publish** ✓

☒ This page is live on the web

<https://bcitcomputing.notion.site/Lab-1-72bdcdf3fac6> Copy web link <sup>3</sup>

Link expires

Never ▾

Allow editing

1-> ☐

Allow comments

2-> ☐

Allow duplicate as template

☒

Search engine indexing


☐


Unpublish

View site

Lastly, go to the **Activities > Assignments** section of learning hub, click the assignment, and then paste the notion link you copied into the text box. Click Submit.

Text Submission





Submit

Cancel

**Failing to follow these instructions will result in no credit being awarded for the lab.**

Question 1A:



The purpose of a Web Server is to take a request from a user's browser for some file (ex - index.html or css file or a picture or pdf), look for that file on the server's hard drive, and once it finds the file, send that file back to the user. These next couple questions are going to give you practice working with files and directories.

For this question, I want you to make a program which prints a list of files in a given directory, filtered by the extension of the files. The path to a directory should be passed as the first argument to your program (e.g. '/path/to/dir/') and a file extension to filter by as the second argument.

As an example, if I pass 'txt' as the second argument then you will need to filter the list of files in the directory to only files that end with .txt.

The list of files should be printed to the console, one file per line.

## Hints:

The `fs.readdir()` method takes a pathname as its first argument and a callback as its second. The callback signature is:

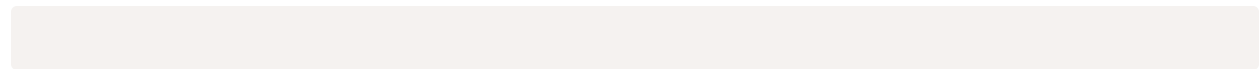
```
function callback (err, list) { /* ... */ }
```

where `list` is an array of filename strings.

Documentation on the `fs` module can be found by pointing your browser here: <https://nodejs.org/api/fs.html>

You may also find node's `path` module helpful, particularly the `extname` method.

Documentation on the `path` module can be found by pointing your browser here: <https://nodejs.org/api/path.html>



Question 1B:

## Problem Breakdown:

Same question as the previous but using modules this time around. You will need to create two files to solve this.

Make a program that prints a list of files in a given directory, filtered by the extension of the files. The first argument is the directory name and the second argument is the extension filter.

Print the list of files (one file per line) to the console. **(Nothing new here, you did this in the previous question).**

You need to write a module file to do most of the work. The module must export a single function that takes three arguments: the directory name, the filename extension string and a callback function, in that order. The filename extension argument must be the same as was passed to your program. i.e. don't turn it into a regular expression or prefix it with "." or do anything else but pass it to your module where you can do what you need to make your filter work.

The callback function must be called using the idiomatic `node(err, data)` convention. This convention states that unless there's an error, the first argument passed to the callback will be null, and the second will be your data. In this case, the data will be your filtered list of files, as an Array. If you receive an error, e.g. from your call to `fs.readdir()`, the callback must be called with the error, and only the error, as the first argument.

You must not print directly to the console from your module file, only from your original program.

In the case of an error bubbling up to your original program file, simply check for it and print an informative message to the console.

**How i'll grade you:** these four things are the contract that your module must follow.

1. Export a single function that takes exactly the arguments described

2. Call the callback with an error or some data as described.
3. Don't change anything else, like global variables or stdout.
4. Handle all the errors that may occur and pass them to the callback.

(You can add in Notion as many blocks of code as you need)