

CS2040 Lab 1

Java Introduction

Lab Introduction

- 2-hour session every Friday
- Intended to provide hands-on experience with programming
- Involves one-day assignments, and take-home assignments
 - One-day assignments (solving 1 problem) should ideally be completed within the lab itself. The actual duration of the deadline is **8am on Friday to 8am on Saturday**, but the assignment is “*doable*” within the duration of the lab.
 - Take-home assignments (solving 2 problems) should be completed before the deadline (2 weeks per take-home assignment). Released on **Tuesdays 10am** of the week they are to be released.
- Slides will be accessible in Canvas under “Files > Labs” at 6pm on Fridays, along with some hint slides for the one-day assignment

Lab Structure

- 1st part of labs (max. 30 min): short lesson
 - Cover answers for the previous session's assignments (one-day, and take-home, where applicable)
 - Cover relevant Java API for the session's given topic
 - Brief discussion of the one-day assignment
- 2nd part of labs: solving the one-day assignment (graded)
 - The first 20-30 minutes of this part will be used for students to plan how to solve the assignment, and express it in terms of pseudocode
 - Allowed to discuss at algorithm level (high level English description or drawing) with other students, but no discussing/sharing of code
 - **Type / write out the more detailed pseudo-code for your solution (should be done individually) which must be submitted to the TAs**
 - Submission of the pseudo-code will count as your attendance for that week's lab
 - Can continue working on the assignment after the lab if necessary

Kattis Introduction

- Online platform used for submitting and grading assignments
- Found at <https://nus.kattis.com/>

*Everyone should have already signed up. If you have not please sign up, refer to lecture notes 0 on how to sign up (the registration key should have been sent to your canvas mail box).

Lab 1 – Runtime Analysis

- In CS2040, sometimes having a program/algorithm give the correct answer may not be sufficient; it may be required that your program/algorithm should run quickly as well
- Kattis provides the maximum running time for a program in the problem description on the right, under “CPU Time limit”
- Generally, machines on Kattis can run close to 100 million operations a second
- Try calculating whether your program can pass the time limit based off its time complexity
- Eg for a problem where $N \leq 1000$, and the time limit is 1 sec, $O(N^3) = 1$ billion = probably not okay (> 100 million), whereas $O(N^2) = 1$ million = okay

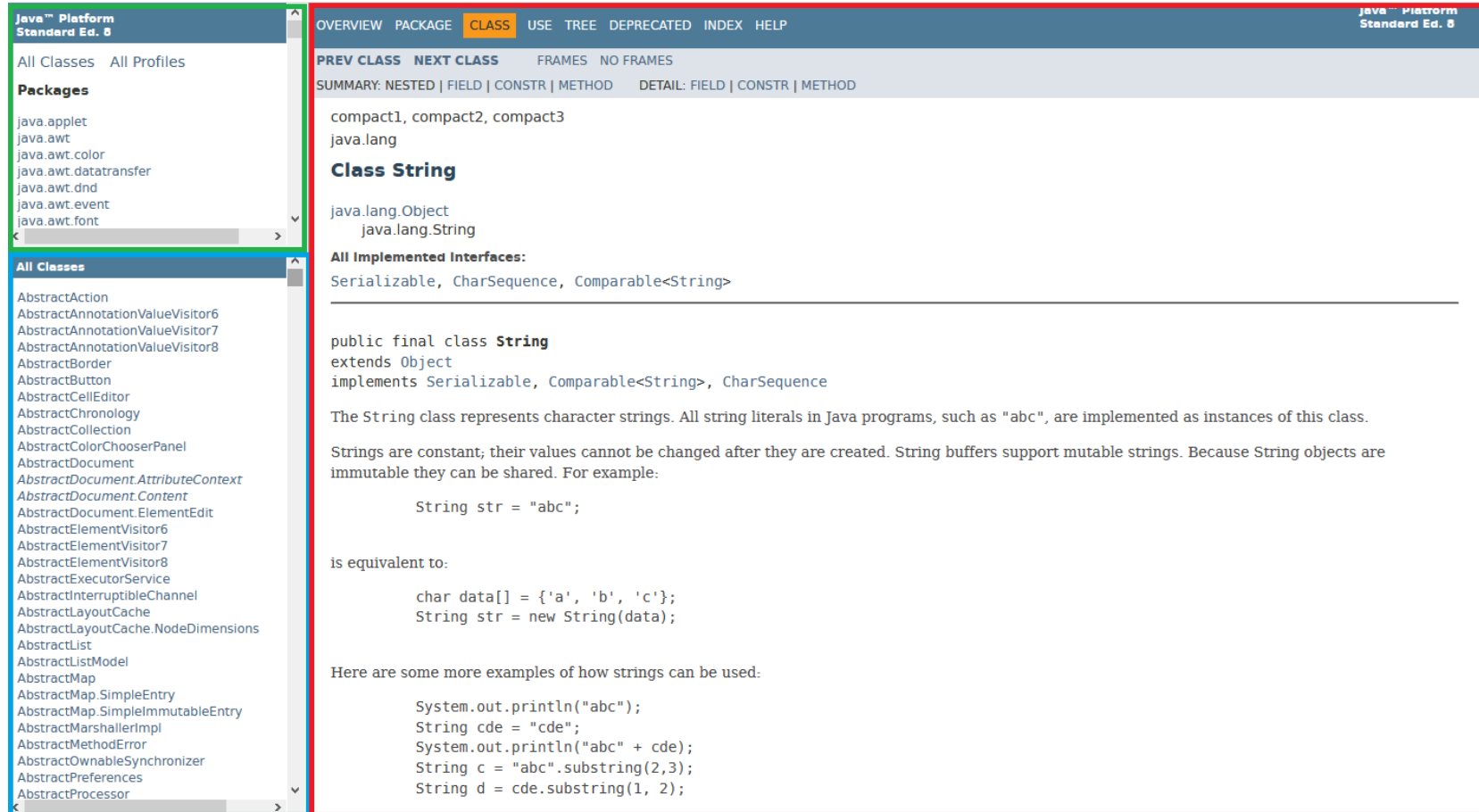
Lab 1 – Runtime Analysis

- Also note that the methods provided by Java API may not run in $O(1)$ time, and as such your program may run slower than intended if this is not taken into consideration
- To fully understand the methods provided by the API, it is recommended to read up the documentation
- The following slides are based off the Java 8 API documentation:
<https://docs.oracle.com/javase/8/docs/api/>
 - Documentation of the later versions of Java (up to Java 10) follows the same idea, but uses a different layout by default. To view them in the same manner, click on “FRAMES” at the top of the page
 - Documentation from Java 11 onwards does not seem to support frames, requiring manual navigation of the packages (or just search for the class name via Google, or the search bar at the upper right of the Java API page)

Lab 1 – Reading Java API

Package window: can click on them to view only classes from a single package

Class window: click on one of them to view the full API of that class



Main window: shows all the details of the selected class

Lab 1 – Reading Java API

- For the package window, clicking on a package will switch the class window to show only the classes in that package
 - Frequently used packages are:
 - `java.util`
 - Almost everything that's covered later in the module
 - `java.lang`
 - Strings and related classes
 - Wrapper classes
 - `java.io`
 - Buffered input/output (covered in a later lab)
 - For Java 9 onwards, these are found under the `java.base` package in the documentation (but retain the same package names otherwise)

Lab 1 – Reading Java API

- For the main window, the contents tend to be sorted as follows:
 - 1. Brief description of class
 - 2. Fields, constructors and methods (short version, sorted alphabetically)
 - Click on an entry here to go to the relevant entry in section 3
 - 3. Fields, constructors and methods (long version, may not be sorted)
- The time complexity of a method will sometimes be stated in section 1 or 3

Lab 1 – Useful API

- Scanner class – used for reading input
- Found in the java.util package; need to use the following line to import:
 - `import java.util.*;`
- Declare a new Scanner object with the following line (in main method):
 - `Scanner sc = new Scanner(System.in);`
 - Do not create more than 1 Scanner object in your program, as using multiple Scanner objects can cause problems
- Read in input using the methods found in Scanner:
 - `int testCases = sc.nextInt();`

Lab 1 – Scanner

Method name	Description	Time
<code>.nextInt()</code>	Reads the next token in the input as an integer	$O(n)$
<code>.nextDouble()</code>	Reads the next token in the input as a double	$O(n)$
<code>.next()</code>	Reads the next token in the input as a String	$O(n)$
<code>.nextLine()</code>	Reads until it reaches the end of the line	$O(n)$

n refers to the length of the input that is read

These slides covering API will cover the most frequently used (but not all) methods of a class; for a full list, refer to the Java API documentation

Lab 1 – System.out (not really a class)

Method name	Description	Time
<code>System.out.print(String str)</code>	Prints <i>str</i>	$O(n)$
<code>System.out.println(String str)</code>	Prints <i>str</i> , followed by a newline character (<code>'\n'</code>)	$O(n)$
<code>System.out.printf(String format, Object... args)</code>	Emulates the printf function from C	$O(n)$

n is the length of the output

Technical note: `System.out` is an instance of the `PrintStream` class, so you may refer to the API documentation on `PrintStream` (in `java.io`) to explore more methods

Lab 1 – Useful API

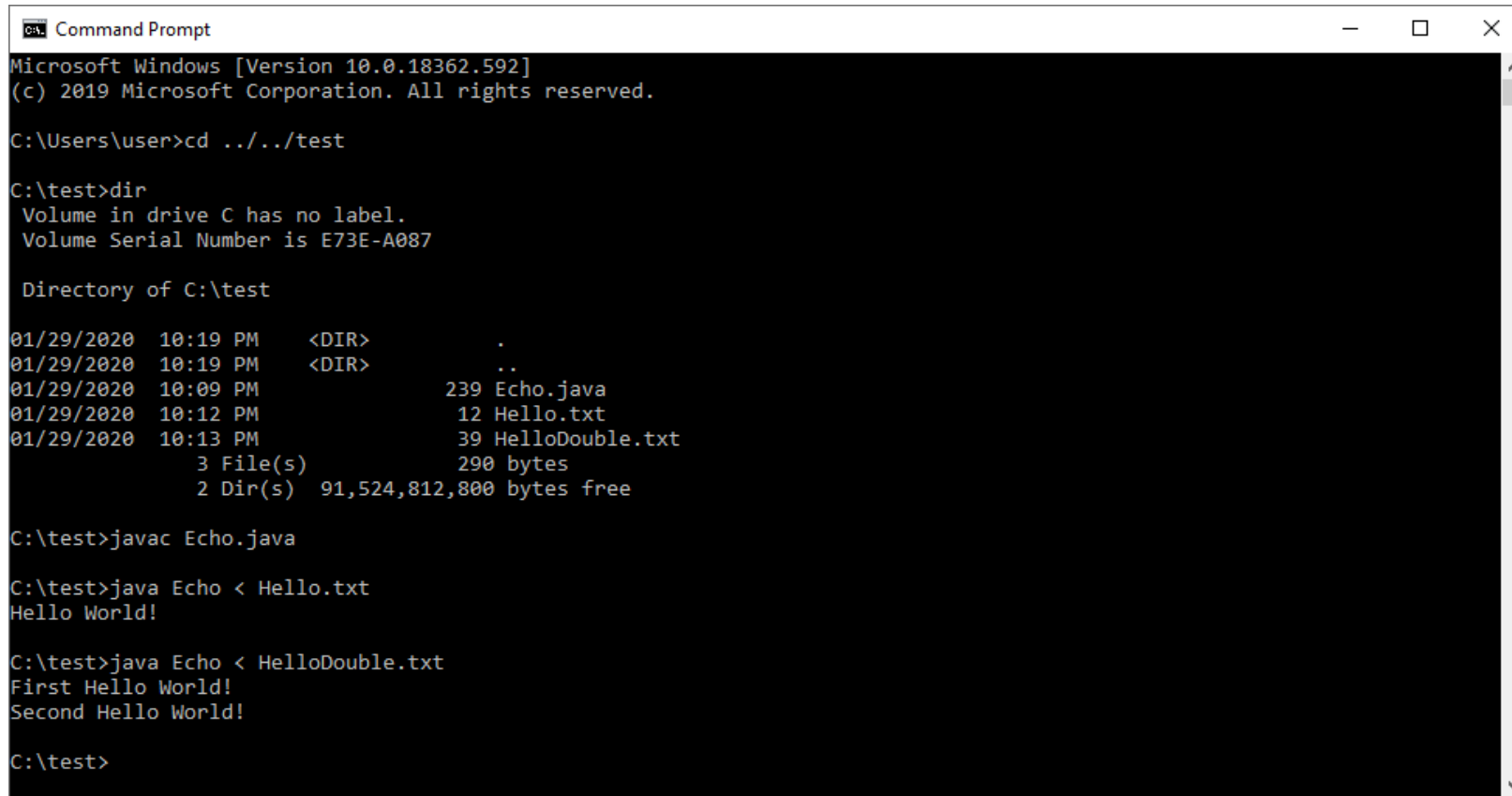
- String class – contains several methods that could be of use
- Found in the java.lang package; is imported by default
- Can be constructed in various ways:
 - `String str1 = new String("apple");`
 - `String str2 = "apple";`
 - `char[] arr = {'a', 'p', 'p', 'l', 'e'};`
 - `String str3 = new String(arr);`

Lab 1 – String

Method name	Description	Time
<code>.split(String delim)</code>	Splits a String into an array of Strings, based off the characters found in <i>delim</i>	O(length of string)
<code>.charAt(index i)</code>	Returns the character at index <i>i</i> (0-based)	O(1)
<code>.equals(String other)</code>	Checks if this string has the same value as the value of String <i>other</i> Note: not the same as using <code>==</code>	O(length of shorter string)
<code>.concat(String other)</code>	Returns a new string which is this string + <i>other</i> Note: does not modify the original String (Strings are immutable in Java)	O(length of resulting string)
<code>.length()</code>	Returns the length of the string	O(1)
<code>.substring(int start, int end)</code>	Returns a new string, which contains the content of the original string from index <i>start</i> (inclusive) to index <i>end</i> (exclusive) (indices are 0-based)	O(length of resulting string)

Lab 1 – Command Line/Terminal Usage

- It is helpful to know how to compile and execute Java programs via command line (example below)



```
Command Prompt
Microsoft Windows [Version 10.0.18362.592]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\user>cd ../../test

C:\test>dir
Volume in drive C has no label.
Volume Serial Number is E73E-A087

Directory of C:\test

01/29/2020  10:19 PM    <DIR>          .
01/29/2020  10:19 PM    <DIR>          ..
01/29/2020  10:09 PM                239 Echo.java
01/29/2020  10:12 PM                 12 Hello.txt
01/29/2020  10:13 PM                 39 HelloDouble.txt
               3 File(s)                290 bytes
               2 Dir(s)  91,524,812,800 bytes free

C:\test>javac Echo.java

C:\test>java Echo < Hello.txt
Hello World!

C:\test>java Echo < HelloDouble.txt
First Hello World!
Second Hello World!

C:\test>
```

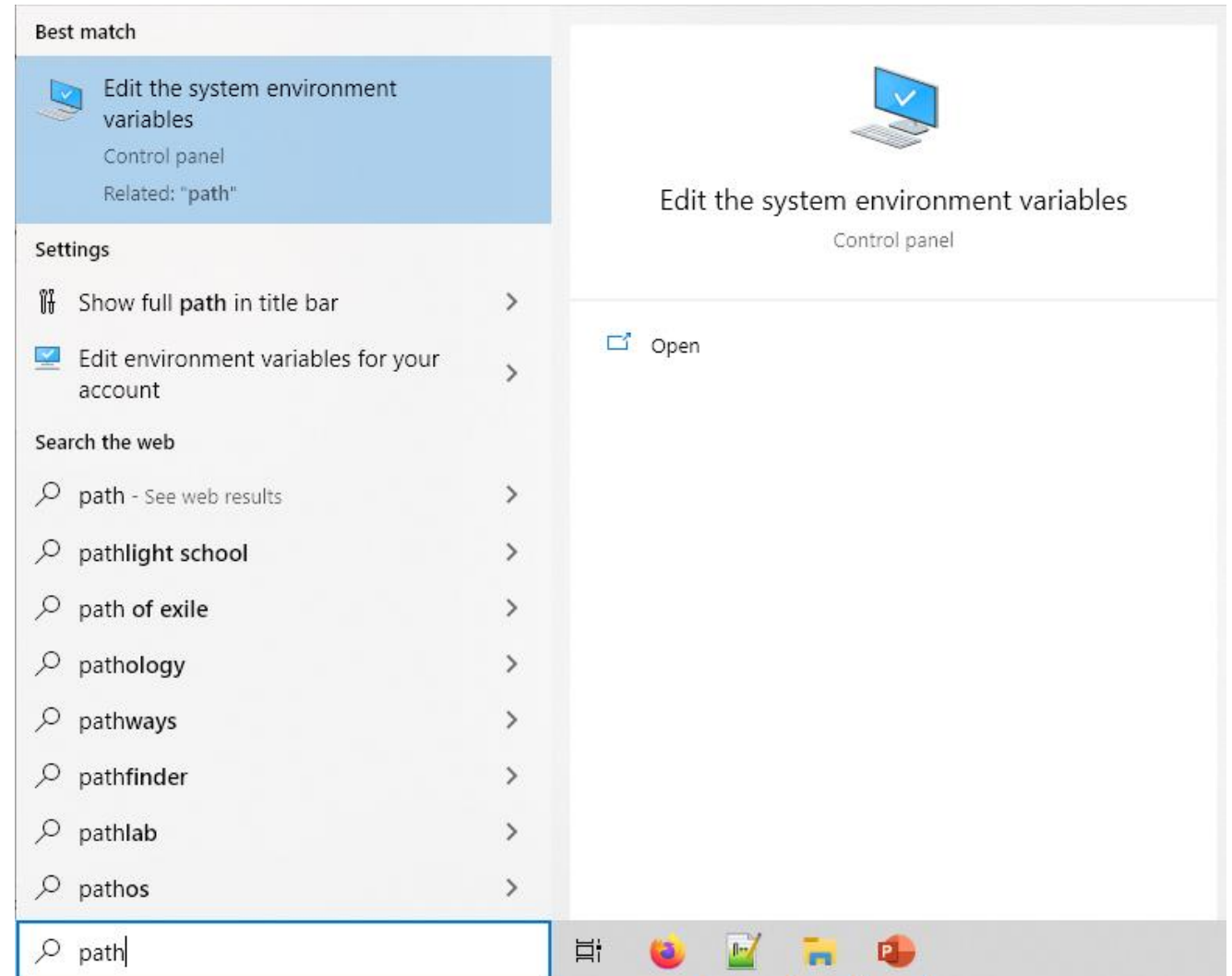
Lab 1 – Command Line/Terminal Usage

- Using the command line requires setting up the location of your java.exe/javac.exe files in your PATH environment variable first, otherwise the command line will be unable to recognise "javac" and "java" as valid commands
- The following slides can also be found as a standalone document on Canvas in the Lectures > Java Intro and ADT folder

Editing PATH Environment Variable

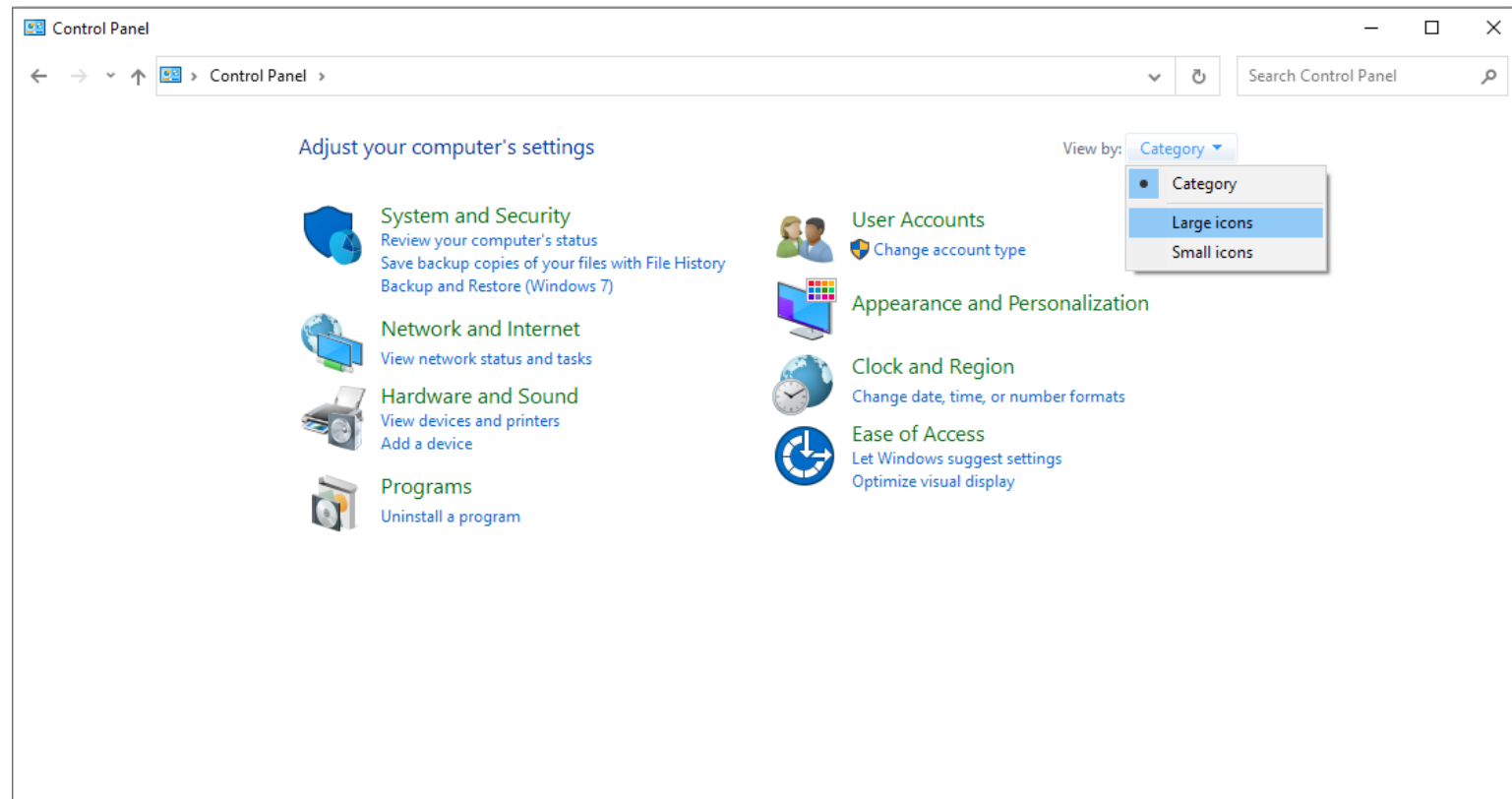
If you have the Windows search bar active, then accessing environment variables can be easily done by searching for "path" and clicking on the first search result

Skip to slide 20 if you do this



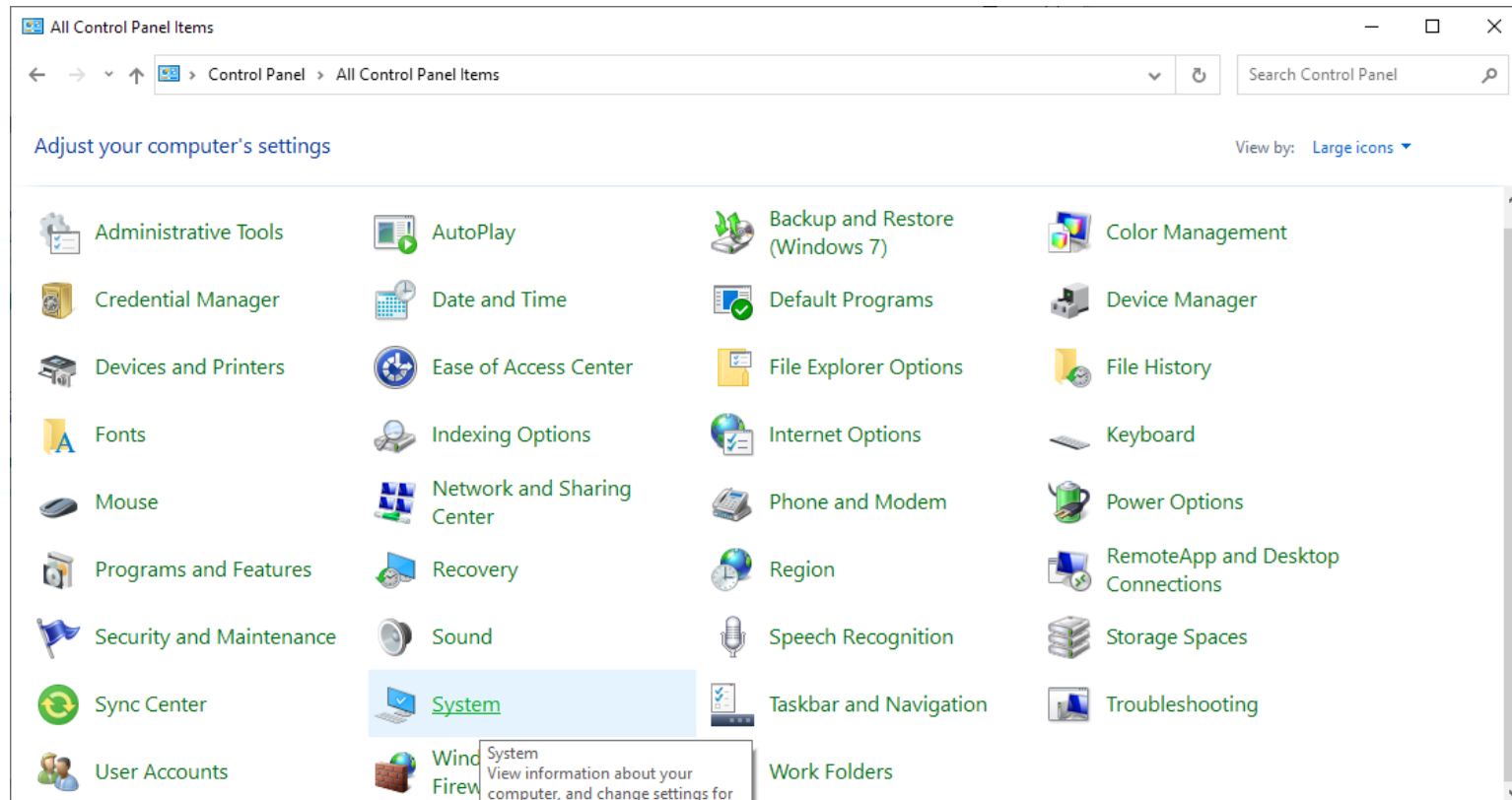
Editing PATH Environment Variable

Otherwise, open the control panel and set "View by:" to "Large icons"



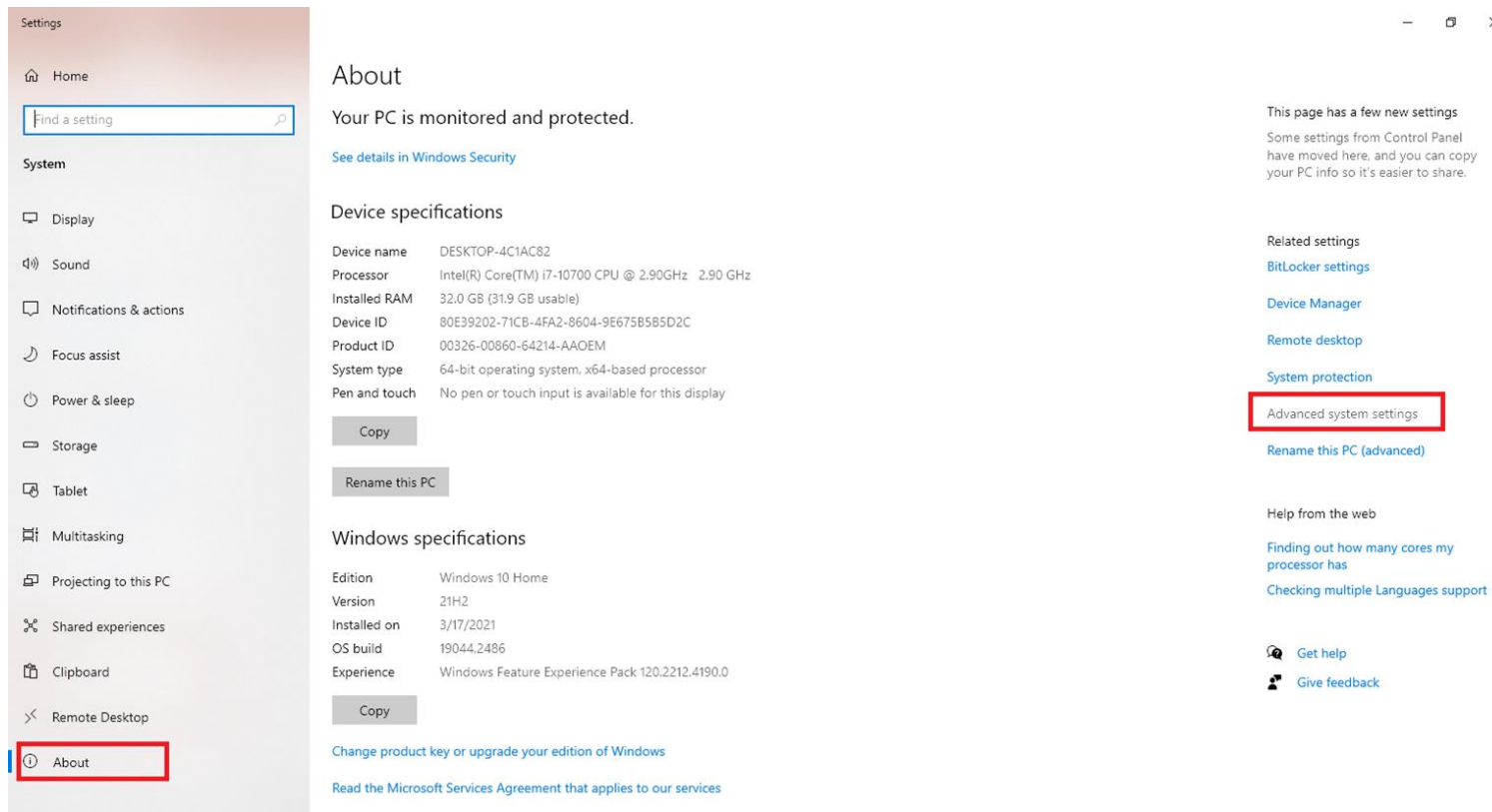
Editing PATH Environment Variable

Click on "System"



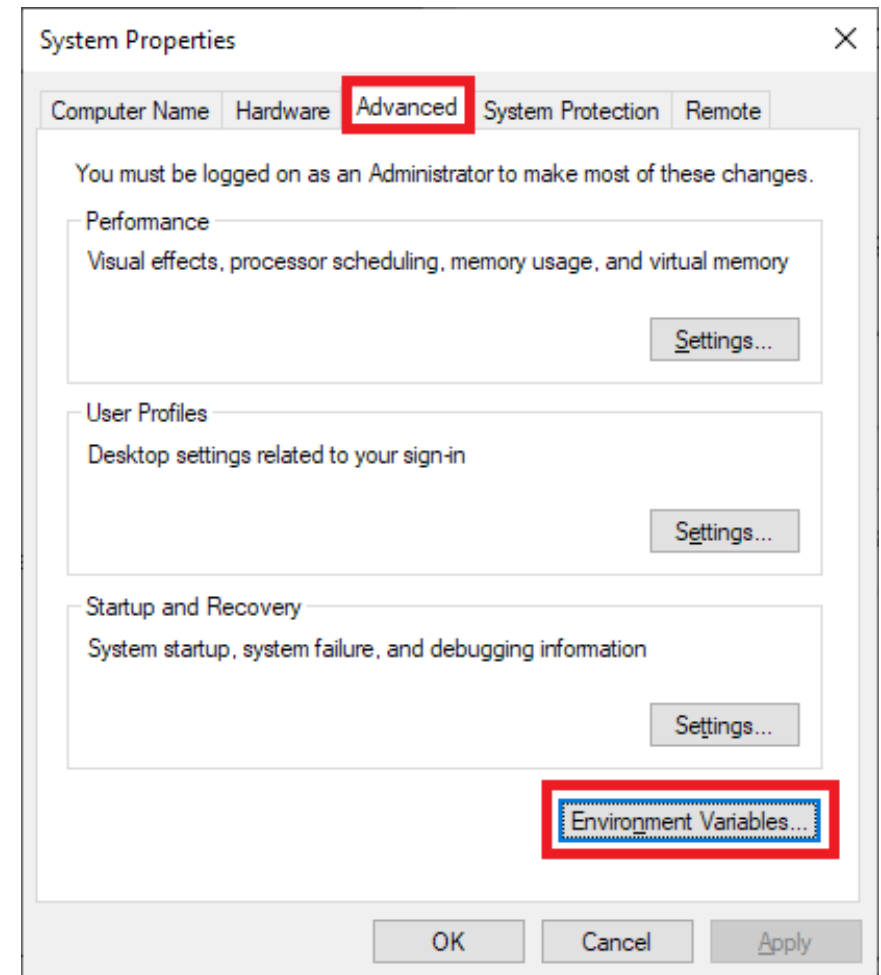
Editing PATH Environment Variable

On the "About" page (should be the default), click on "Advanced system settings" on the right



Editing PATH Environment Variable

Go to the "Advanced" tab (again, should be the default) and click on "Environment Variables..." at the bottom of the menu



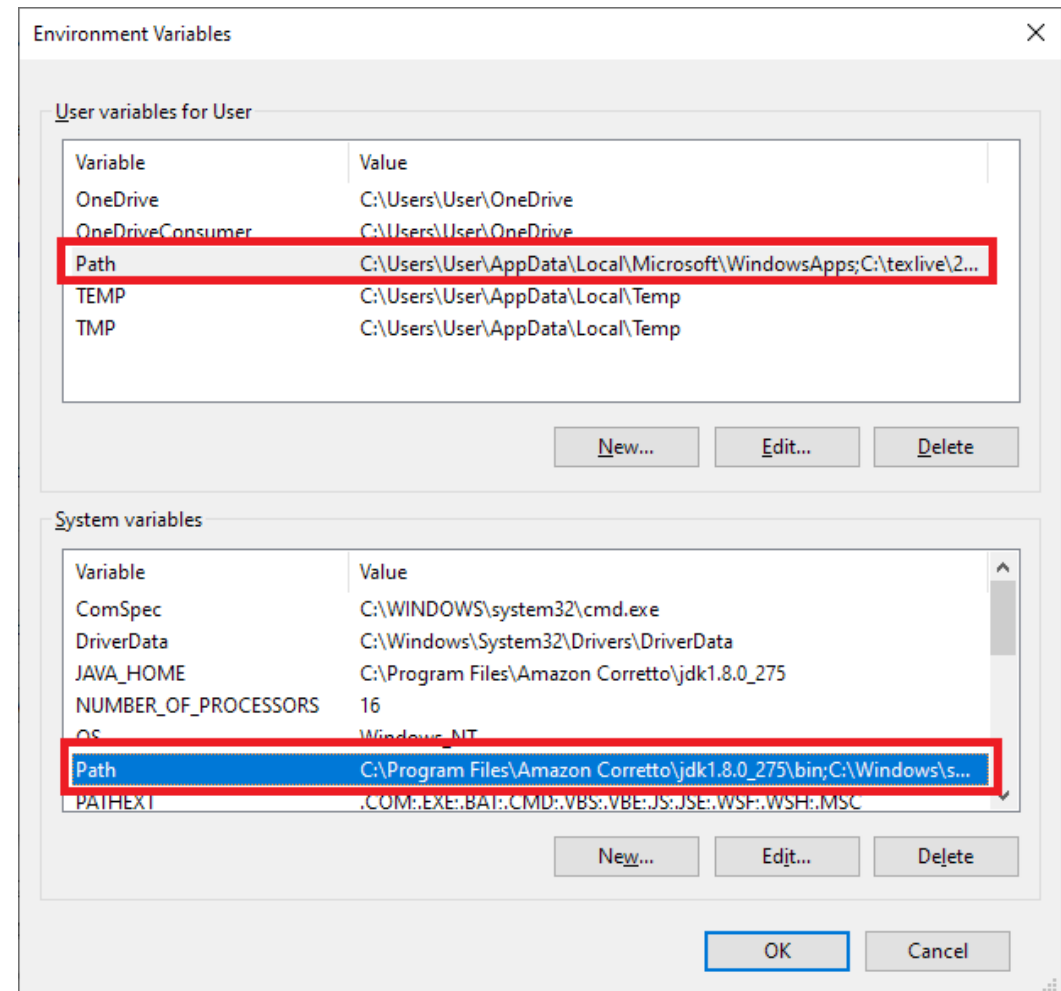
Editing PATH Environment Variable

From the following menu, you will find 2 different "Path" variables; editing either is fine.

Editing the "User" variables will only affect the current user.

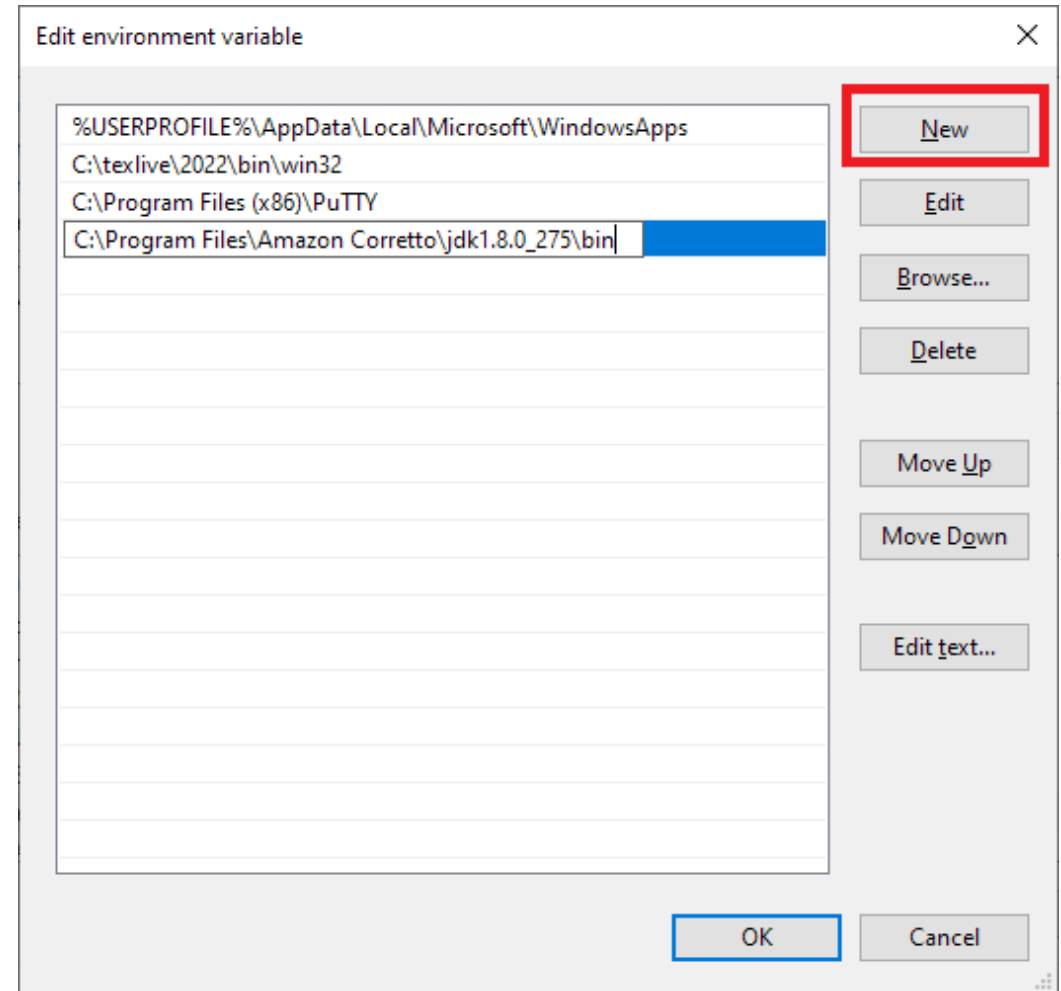
Editing the "System" variables will affect all users.

Editing can be done by double clicking that row.



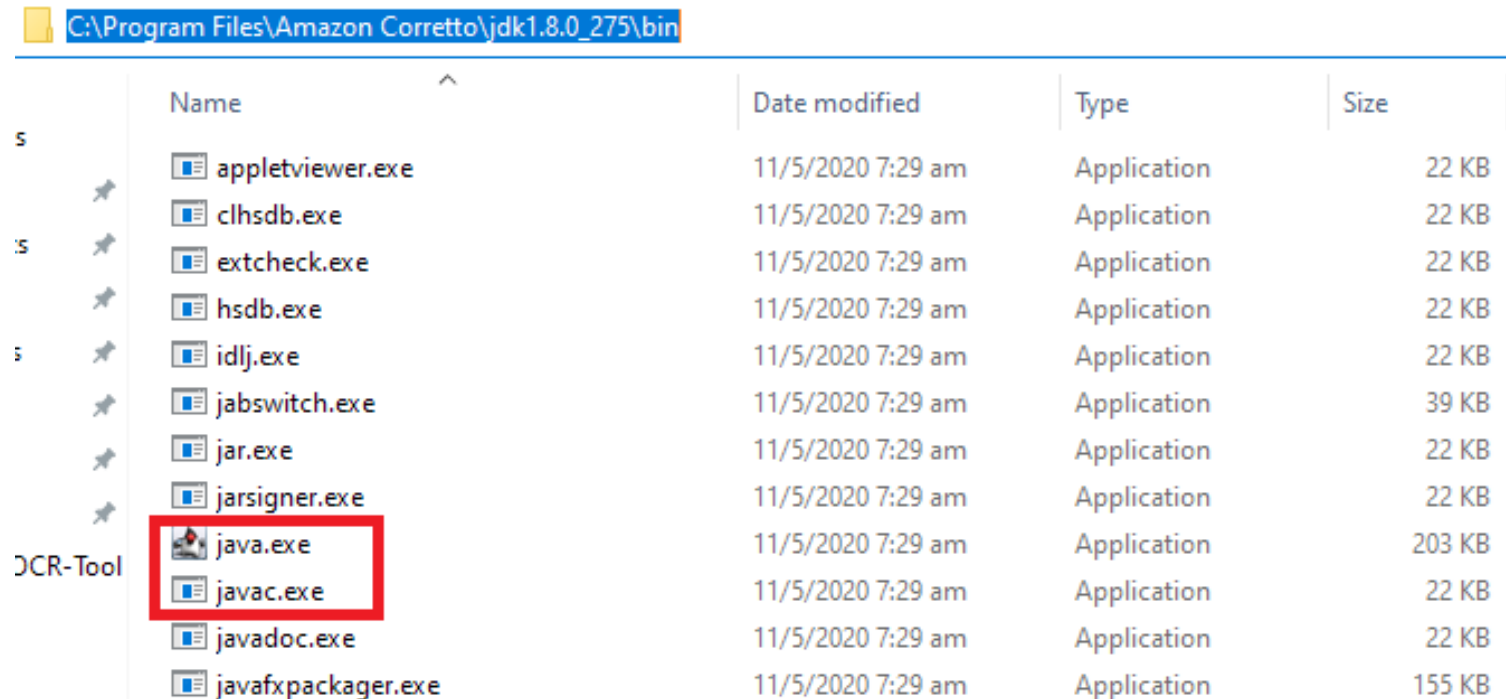
Editing PATH Environment Variable

In the window that follows, select "New" and enter the path of the folder containing "java.exe" and "javac.exe" for your device



Editing PATH Environment Variable

For this example, "java.exe" and "javac.exe" are found at "C:\Program Files\Amazon Corretto\jdk1.8.0_275\bin", but it may be different on your device



C:\Program Files\Amazon Corretto\jdk1.8.0_275\bin				
	Name	Date modified	Type	Size
	appletviewer.exe	11/5/2020 7:29 am	Application	22 KB
	clhsdb.exe	11/5/2020 7:29 am	Application	22 KB
	extcheck.exe	11/5/2020 7:29 am	Application	22 KB
	hsdb.exe	11/5/2020 7:29 am	Application	22 KB
	idlj.exe	11/5/2020 7:29 am	Application	22 KB
	jabswitch.exe	11/5/2020 7:29 am	Application	39 KB
	jar.exe	11/5/2020 7:29 am	Application	22 KB
	jarsigner.exe	11/5/2020 7:29 am	Application	22 KB
	java.exe	11/5/2020 7:29 am	Application	203 KB
	javadoc.exe	11/5/2020 7:29 am	Application	22 KB
	javafxpackager.exe	11/5/2020 7:29 am	Application	155 KB

Lab 1 – Command Line/Terminal Usage

- `cd ../../test`
 - “cd” is the command used to move between folders. “..” means to go up one folder
- `dir`
 - Used to list the contents of a folder. Use “/s” (lowercase L, lowercase S) on Unix-like systems or Powershell
- `javac Echo.java`
 - “javac” is the command use to compile Java programs. The full file name (inclusive of “.java”) is required, and seeing no output is a **good** thing.
 - Each class in the Java file is compiled to its own “.class” file

Lab 1 – Command Line/Terminal Usage

- `java Echo < Hello.txt`
- `java Echo < HelloDouble.txt`
 - “java” is the command used to run compiled Java programs. No filename extension (.java, .class) should be provided
 - “<” followed by the name of a file is used to provide input from a file (known as input redirection).
Instead of having to type out your input manually; simply save your input in a file and use “<” to avoid retyping input each time you test your program
- `java Echo > Bye.txt`
 - “>” followed by the name of a file is used to send output to the file (known as output redirection)
 - Can be used together with input redirection, e.g. *java Echo < Hello.txt > Bye.txt*

One-Day Assignment 0 – Pea Soup

- First one-day assignment is ungraded
- Found on the main Kattis course page at https://nus.kattis.com/courses/CS2040/CS2040_S1_AY2425.
- Writing pseudocode for this assignment is not necessary, but will be required from the next one-day assignment onwards

One-Day Assignment 0 – Pea Soup

- Given a list of restaurants, and their menus, determine the first restaurant appearing in the list that offers both “pea soup” and “pancakes”
- Should be the exact strings “pea soup” and “pancakes”; even if a menu item contains the substring “pea soup” or “pancakes”, ignore it (see Sample Input 2)

Assignment Guidelines

- **Important:** Include your **name** and **student number** in comments at the top of your code

Rehash of “Rules for CS2040 Assignments”:

- If you discuss the problem with any other student(s), include their name(s) as collaborators in a comment at the top of your code
- No usage of anyone else’s code (outside of code provided in lecture materials)
 - Directly using (eg. copy-paste) is not allowed
 - Using code as a reference is not allowed either
 - No AI Tools allowed even for generating algorithm/pseudo-code only.