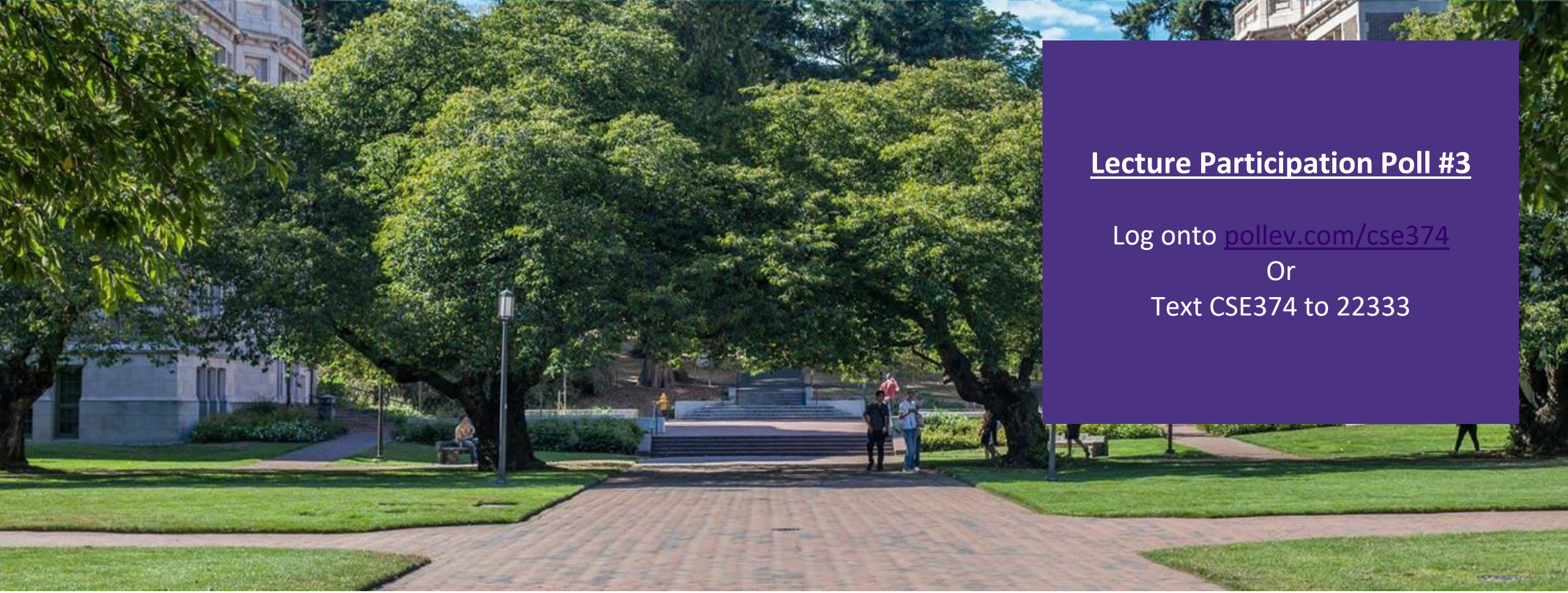

Lecture Participation Poll #3

Log onto pollev.com/cse374
Or
Text CSE374 to 22333

Lecture 3: Bash Basics

CSE 374: Intermediate Programming Concepts and Tools

Administrivia

HW 1 will be posted later today

- will post instructions on setup and HW to course webpage
  - will include access to Klaatu Linux accounts
- will announce posting via Ed board (which sends emails)

Sorry about Canvas weirdness…

- Working with UW IT to fix panopto and zoom tabs
  - Lecture 2 video is now on Panopto Recordings tab of Canvas
- Going forward goal is to have lectures recorded and uploaded as well as a live sign in link, but figuring that out

Shell State

- Each new instance of the bash shell maintains a "state"
  - Current location in the file system
- PATH variable
  - Whenever a command is executed there is a list of pre-defined locations bash looks
  - PATH holds the list of pre-designed directories
  - echo $PATH
- Bash rc file
  - A shell script that is automatically run whenever Bash starts up
  - Used to initialize your shell state
  - You can find example files shard online
  - Exists locally on your machine, not on the remote linux server
- Bash history file
  - .bash_history is a hidden file that automatically logs your command history

https://astrobiomike.github.io/unix/modifying_your_path

https://www.gnu.org/software/bash/manual/html_node/Bash-Startup-Files.html

Shell Variables

- Shell variables = string substitution
  - Declare variables in the shell to easily refer to a given string
  - All variables are strings
- Declare variables in the terminal with a name and a string value
  - `<var name>="<var string>"`
  - EX: `myvar="myvalue"`
    - Note: no white space allowed on either side of the "="
- Refer to your variable using the "`$`" symbol before the var name
  - `$<var name>`
  - EX: `echo $myvar`
    - `myvalue`
- Alias
  - Rename a bash command, create your own shortcut
  - `alias <string>="substitution string"`
    - EX: `alias cheer="echo hip hip horray!"`
  - Only exists within the currents state of your shell
  - Can store alias in bashrc file to preserve alias across all shells

Bash Script Variables

- When writing scripts you can use the following default variables

$# - stores number of parameters entered

Ex: `if [$# -lt 1]` tests if script was passed less than 1 argument

$N - returns Nth argument passed to script

Ex: `sort $1` passes first string passed into script into sort command

$0 – command name

Ex: `echo "$0 needs 1 argument"` prints "<name of script> needs 1 argument"

$* returns all arguments

$@ returns a space separated string containing all arguments

"$@" prevents args originally quoted from being read as multiple args

Alias

- defines a shortcut or ‘alias’ to a command
  - `alias <string>=“substitution string”`
    - EX: `alias cheer=“echo hip hip horray!”`
  - technically just string replacement
  - Only exists within the currents state of your shell
  - Can store alias in bashrc file to preserve alias across all shells

Special Variables

$HOME – sets home directory

EX: $HOME=~/CSE374 would reset your home directory to always be CSE374

$PS1 – sets prompt

$PATH – tells shell where to look for things

printenv shows current state

Math in Bash

- Everything is interpreted as a string
  - No concept of integer values
- To do math use double parentheses to interpret as arithmetic expression
  - `RES=$((1+2))`
    - `$RES` will be set to "3"
  - `MYVAR=$((RES+2))`
    - `$MYVAR` will be "5"
  - `K=$i + $j does not add numbers`
    - `k=$(($i+$j))` will add numbers

Quotes in the Shell

- Double quotes can be used to wrap a string with white space into a single argument
  - EX: `myvar="Some string"`
- Single quotes tell the shell to treat the string as a literal
  - No variable expansion or command substitution
  - EX: `echo '$myvar'`
    - `$myvar`
    - `echo "$myvar"`
    - `Some string`

Demo: Variables and Aliases