**Table of Contents & Summary of Lectures**

1. Bright Path’s Orientation & Expectations-D1W1
2. Software Development-Day 1, Week 1
3. Software Development(Creating, Designing, Testing, Deploying)
4. Common software methodologies
5. Waterfall
6. Agile
7. Feature-Driven Development
8. Lean
9. Rad

c. Software Stacks (Python, JavaScript, Ruby)

d. Stack(database, web framework (Back-end), Front-end framework, Back- end language/environment/server)

1. MEAN/MERN Stack(Full Stack JavaScript on the back-end and front-end).
2. MongoDB: A NoSQL database
3. Express.js: A web framework
4. Angular.js: A front-end framework/React.js: Another front-end framework
5. Node.js: A back-end “language”/server
6. Python Stack
7. Flask or Django: Web frameworks
8. React.js: A front-end framework
9. PostgreSQL: A database
10. Python: Back-end language

iii. Roles in Software

1. Engineering Roles
2. Web Developer(Full-Stack, Front-end, Back-end)
3. Mobile Developer(Android, IOS, hybrid platforms)
4. Software Developer
5. DevOps Engineer
6. QA Engineer
7. Security Engineer
8. The Software Team
9. Project Manager
10. Engineer Manager
11. Tech Lead
12. Business Analyst
13. Developers/Engineers
14. Architect
15. Other Roles
16. Data Analytics-Work with business or engineering to make data-driven decisions.
17. Developer Operations/Production Engineering-ensure that code is successfully deployed & installed. Ensure code stays working on the public-facing side of the product.
18. Security Engineering-Build tools and enforce best practices around data privacy. Ensure systems are protected against malicious actors, etc.
19. Running Code-Day 2, Week 1
20. Syntax, Syntax recall
21. Command Line
22. GUI-Graphical User Interface(something you can point & click. I.E. File Explorer or Finder)
23. ls: show me all the files/folders where I am
24. cd: go somewhere else
25. ~ : means “this user’s home directory”
26. cd ~ : means go back to home directory
27. cd.. :means the directory above where I am
28. Pwd: tell me where I am in the filesystem
29. mv: more or rename a file
30. To Move:
31. need two arguments for mv
32. To Move-what to move, and where to move it
33. To Move-$ mv somefile ~/Downloads
34. To Rename files:
35. need two arguments for mv
36. What to rename, and the new name
37. $ mv somefile anothername
38. Create new files
39. Need one argument for “touch”: name of file
40. $ touch mynewfile
41. Print File
42. Need one argument for “cat”: which file to print
43. Can provide name of file in current directory
44. Can also provide URL
45. $ cat somefile
46. Tab Completion
47. Press tab when done & it could autocomplete
48. Git
49. A version control.
50. Allows to track changes, prevent bugs, & loss of projects.
51. Git repository
52. A repository is a code project.
53. Contained in single folder (aka directory)
54. Git Basics
55. git init : initialize a local repository right here in this folder
56. git add : add some files to be tracked
57. git commit : create a snapshot of files (needs -m +”message”)
58. git status : tell me about this repo
59. git log : show me a list of the commits, in reverse order
60. git diff : what have I changed since last commit
61. Variables & Datatypes
62. Variable Declaration(var, let, const)
63. Datatypes:
64. Number = 7
65. String = “hello”
66. Boolean = true
67. Undefined = undefined
68. Null = null
69. Array : []
70. Object = {}
71. Mathematical Operators
72. + : addition
73. - : subtraction
74. \* : multiplication
75. / : division
76. % : modulo (remainder) -only returns the remainder value
77. Running JavaScript
78. Getting User Input
79. Basic “If” Statements

If() {

console.log();

} else {

console.log()

}

1. Making Decisions 1-Day 3, Week 1
2. Conditional Logical Operators
3. If Statements
4. If Else Statements
5. If, Else, Else if
6. Loops
7. Pair Programming
8. Github
9. Making Decisions 2-Day 4, Week 1
10. Agile Theory-Day 5, Week 1
11. Functions 1-Day 6, Week 2
12. Object Orientation-Day 7, Week 2
13. Functions 2-Day 8, Week 2

10. Design-Day 9, Week 2

11. Object Orientation Theory-Day 10, Week 2

12. HTML & CSS 1-Day 11, Week 3

13. How The Web Works-Day 12, Week 3

14. HTML & CSS 2-Day 13, Week 3

15. Interactive Web Pages-Day 14, Week 3

16. Interactive Web Pages 2-Day 15, Week 3

17. Back End 1-Day 16, Week 4

18. Back End 2-Day 17, Week 4