Caroline is the best

Goals are good

* + Intro to programming language
  + Differences between languages
  + Programming environments
    - Other the hardward -> different compiler languages (what is a compiler)
    - Pycharm is a IDE
  + Start programming
* Always think about the future
  + Grasp fundamentals
  + Apply concepts to other langauges
* What is a programming language
  + They automate and execute algorithms in structured format
    - Algorithm = set of instructions to solve a problem
  + Some algorithms that cannot be solved using an algorithm
    - Undecidable – what to wear tomorrow
    - Unsolvable – math has produced no solid proof for
    - Intractable – (our focus) – problem that can be theoretical be solved but may take may too many resources
      * Get into algorithmic efficiency
  + Programming languages are intermediary between natural lang and assembly lang
* How does a programming language compare to natural language
  + Natural language is “human” language (ie English)
  + PL is defined by syntax and semantics (lexicon of keywords)
  + PL is unambiguous -> computer unable to understand ambiguity
  + Implementable
  + Turing complete \*\*
    - Named from Allen Turing
    - A turning machine -> represents a set of finite set of instructions
    - There exists a result R such that all machines T and I will produce the same i
    - Ie arithmetic 2 +2 always yield 4
* Your computer system operates using micro-operating instructions
  + There is a lot more going on that the programming language guides you through
* Types of programming language
  + High level – more similar to natural lang
    - Hand computational problems designed by programmer
    - Constructs represent math
    - Such as C++, java, python, MATLAB, C# (web devo language), javascript ‘
    - C++ -> has data structure of high level language/ computational strong and the capabilities to access the low level language
  + Low level (assembly and machine language that computer hardware uses)
    - Direct manipulate the hardware and manipulate the micro-architecture comp uses
* Which language should I choose?
  + Functionality -> what are you going to be using it for
    - Data analysis, machine learning -> python, Matlab
    - Software devo and engineering -> java and C++
    - Hardware manipulation -> C
    - Database management, query -> SQL, MySQL, PostGres
    - Web development -> c#, javascript, html (inspect element)
  + What is machine learning (david)?
* Why python
  + Functionality -> fundamentals of computer science made easier by python
  + Machine learning libraries for data science projects
    - Libraries -> are tools that can communicate with your language
    - You can formfit the functionality of what you are doing with the language to make more specific to what you are doing
  + High demand, used widely
  + Open-source, free of cost
  + Extensively documented
    - Important for troubleshooting
* PROGRAMMING ENVIRONMENTS
* IDE – integrated development environment -> support a few select program per IDE
  + Use pycharm, supports python
  + Netbeans and intelliJ support java
  + Xcode built for mac (java, c++)
* Organization
  + One project will often have more than one program for modularity purposes
  + Multiple projects can be organized in same IDE
  + Debugging
    - Most important
    - No one write perfect program in first pass -> most time spent debugging to understand why what you wrote is not working
    - Breakpoints: allow you to see which value are beign changed as the program is being executed
    - IDE VS PLATFORM (TD)
      * What does it mean for IDE to be on top of Platform

Compiler

* + Each language will have multiple version of a compiler
  + Check syntax
    - Basically a spell checker for you -> makes sure that program is written correctly before you execute/run it
    - Throws in a compiler error before program is executed
  + Translates high level programs into machine-readable format
* Saving and reproduce code
  + Always save your code
    - Cloud serves -> dropbox
    - Google drive
  + GitHUb
    - Programmers preferred way of storing, sharing, and viewing code
    - Social media for programmers-> show off code
    - Get an account for next time
  + Personal Zen is in github
    - Store images/ snapshots
    - Will have code and then have snapshots of what user interface show
  + Maybe find a few examples of excellent github accounts
  + What exactly is an IDE -> how to conceptualize it
* OPEN DISCUSSION:
* Which topics are you interested you most?
* Want more in depth?
* Structure class to be most conducive to learning how to code?
* Anything you didn’t understand?
* Handle hw assignments and readings?
  + Have hw assignments
  + Learn concepts and apply in hw -> quizzes (what is wrong with this code and have to correct it)
  + Be able to do the pset without being able to access the resources
    - Alex bring example of hw assignment
  + By end of class – goal was to do an activity so all parts of the class would be additional steps towards goal
    - Buld a house example by end of class -> end all skills needed to build a house
  + Start by program something just outside of SPSS compabilities (ie outlier, moderator/mediator)
    - Ie simple question
* Resources for commonalities between languages ie EXCEL -> SPSS
  + Stackoverflow -> additional resources
* Syntax vs code vs script
  + Syntax – specific to querying language to look through columns of data
    - Subset of code
  + Script – overarching term for writing code (the result)
  + Coding is the writing of the whole thing
  + Note: many different programs differences on what they call similar and different things
* Sarah to email course
* David -> university level class (online class) – Udacity
* Alex – excel
* What is caroline username
* How do you add caroline on github