Stations to explore evolution of languages and products

Goal: Survey of some languages and contexts for which programming is done, to explore:

- What features persist across evolution (and why)
    - Focus on flaws or deficiencies – what was the trade-off where this "negative" thing won out?
    - Focus on banality of core features. What was the trade-off, and why not change this?
- What features change across evolution (and why)
    - Focus on positive attributes and human value associated with them (or estimated value, in the case they didn't persist for long)
    - Think about the pattern from Burden's Wheel: tech that centralizes/networks increases opportunities and decreases cost/increases wages → makes life better; may begin as business/elite only, but ultimately becomes available for all, creeping into all aspects of our lives
- What "low" and "high"-level languages are, based on how they are read/written, and the level of abstraction they have (or lack of abstraction, and inclusion only of primitive data types).

Everyone complete A first!

A. "A Hierarchy of Languages"

1. A high level language is _______________ while a low-level language is ________________.
2. A c__________________ and i__________________ are used to convert a high-level language into a low-level language that can work on a particular computer. This means that when a programmer makes a high-level program, it could work on a wide range of computers.
3. A program in assembly language is (circle one) longer shorter than a program written in a high-level language. This means that a program in assembly language is (circle one) harder easier to read than one written in a high-level language.
4. M__________________ language is the lowest-level language. Each piece of code is in the form of n_________________.
5. You can think of moving from a high level language to a low level language as being a movement from abstraction on the macro level to the micro level. Each program shown (there are three – one in each rectangular space) has their own macro and micro features, but we really lose sight of the macro as you progress to the lower level language. In the example on page 15:
    a. C language code (high-level language):
        i. Macro = Process to draw a _____ including moving a pen and drawing lines
        ii. Micro = (x,y) coordinates of ____________________________________
        The space to draw in ("hdc" in this example)
    b. Assembly Language:
        i. Micro: state of being inside of outside of window (enter or leave), state of some variable/variables related to drawing (si, di, ds, ax), amount of offset in y, amount of offset in x, state of moving or drawing (MoveTo, LineTo), etc.
    c. Machine Language:
        i. Micro: hexidecimals (numbers of base 16) that represent each piece of data/relationships

Groups can move around – complete B, C, D, E, and F in any order.

B. "How Assembly Language Works"

1. "Each instruction in assembly language corresponds to a single instruction for a ____ (______)." This makes Assembly a ____-level language that is relatively hard to read/code.
2. Every different cpu has its own a________________________ l____________________.
3. Registers are types of _______________________ that can be accessed more quickly than the computer's ___________________________________. Because registers handle temporary memory, these may be associated with (circle one) RAM / ROM. A 1 G processor means that the register is ______________ wide.
4. Registers include:

   a. Address registers: containing ________________ locations.
      i. If there are a pile of addresses, then a s__________________ p__________________ register is used to point to the top of the stack (pile).
   b. Instruction _________________ (IP) : location of next ____________________

5. Look at the assembly code in the third column of the last picture on page 176. You can use the description (an English translation of what is going on, by line) to see what each line means. Based on this, you can see the micro elements of assembly code for this example:
   Abstraction (at the macro level): Creating a sum of two numbers.
   Micro: VarA and VarB are _______________ (tell what kind of data), Result is ________________, Place to store VarA, VarB and Result
6. The graphic at the top of page 177 shows the state of the machine changing. Note that the IP is changing. Why (reflect on your answer in question #4, above)?

What number will IP point to when the process is done (it is not 109)? Use the graphic on the prior page (third one) to help.

7. The width of a processor dictates how it can do some math. Part #6 on page 177 explains this. Imagine you had a calculator that only has a processor wide enough to store a 3-digit number (like 123). How would you handle adding two 6-digit numbers (like 123456)? Explain what the calculator would need to do before trying any addition, and then again before the result is revealed. Show with this example on paper:
   123456
   $\underline{+\ \ 1212}$

C. “Programming a Turtle”

1. LOGO is how Mrs. Frazier was first exposed to computer science (in 3rd grade).
   a. Macro: Drawing a shape including moving a pen and drawing
   b. Micro: Pen height (u____ or d______), Direction (r____________ and degrees), Distance f___________ (for example: 10)
2. Note there is something in between a simple macro and micro level: LOGO actually records the sequence of data and instructions as B_____ (a three-letter name). This creates polymorphism. Explain.

3. Visit https://gym.pencilcode.net/draw/#/draw/line.html (Safari may not work; try a different browser) and edit the PencilCode (very similar to LOGO) to write the first letter of your name with at least three line segments. Write the code you used here, along with a sketch of the letter as you drew it:

D. “How the C Language Works” *Yes, there is a missing second page. It’s OK!*

1. C is high level because it can be “recompiled to run it on another _______________________.”
2. C is low level because in C you can directly control m________________ locations including registers associated with the C___. You can call a____________________ l_______________________ functions in C as well.

3. Pointers point to variables to give them initial (starting) values and set them in memory. But beware – if the pointer is not thoroughly established, code will cr______. The paper warns: "Danger: Beware ________________ Pointers."

If you took Java Programming at MV, now you know why your teachers insisted you always declared AND initialized your variables. Skipping this in C → BAD NEWS.

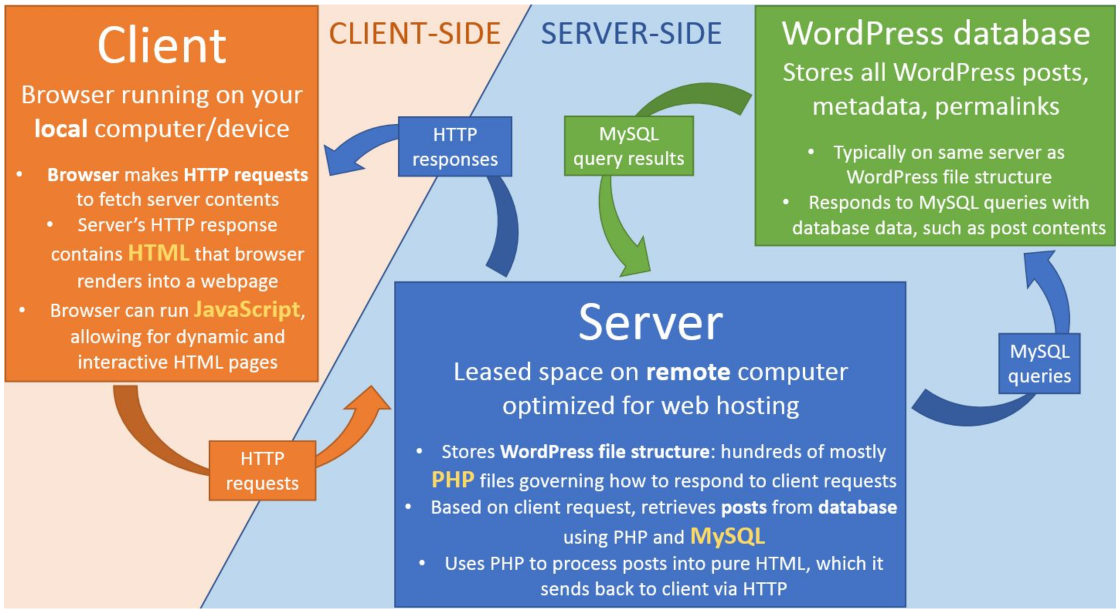
4. Consider the example on page 181.
   a. Macro: Abstraction changes a string (word or sentence) from l__________________ to u__________________________
   b. Micro: Each c______________ (letter), state of character (if lowercase, return the result 1; else return result ___), constant characters to compare to ('a' to '___'), value of a character (zero or another number that is an ASCII value), incrementor variable so can repeat process.
   c. Note the use of logic here. For AND, what symbol is used? _____
   d. There is selection in this example. What keywords indicate decisions/choices? i___ and e________
   e. There is iteration in this example. What is happening over and over?

   What keyword is used to indicate there is iteration? w____________________

E. Look at the mini-posters for Javascript and PHP

1. These languages are in_____________________ languages, meaning a web br_______ reads source code.

2. HTML and PHP are both c___________-side interpreted languages (runs on your ________ computer/device), while PHP is a s____________-side interpreted language (runs on a remote computer with special space/memory for your content). See the graphic below to better understand what this means and complete some of these blanks *(taken from a presentation about how WordPress, a popular website/blog tool, works)*.


Client
Browser running on your local computer/device
Browser makes HTTP requests to fetch server contents
Server's HTTP response contains HTML that browser renders into a webpage
Browser can run JavaScript, allowing for dynamic and interactive HTML pages
CLIENT-SIDE
SERVER-SIDE
HTTP responses
MySQL query results
WordPress database
Stores all WordPress posts, metadata, permalinks
Typically on same server as WordPress file structure
Responds to MySQL queries with database data, such as post contents
MySQL queries
HTTP requests
Server
Leased space on remote computer optimized for web hosting
Stores WordPress file structure: hundreds of mostly PHP files governing how to respond to client requests
Based on client request, retrieves posts from database using PHP and MySQL
Uses PHP to process posts into pure HTML, which it sends back to client via HTTP

3. As these are interpreted languages, they are not com______________, meaning a computer does not process them first and report errors. This can make them hard to troubleshoot/debug.

4. On the PHP Poster: Two other languages are shown here, that PHP can support: H_______, C______.

5. Interpreted languages are great for making lay____ of websites, UI(user interface, aka user interactivity) for websites.

6. On the Javascript Poster: Javascript enables auto_____________, but also allows you to override it and enter text into forms yourself.

F. Look at the mini-posters for Python, Java, C++, and Objective C (~ Swift)

1. All of these are Obj_______________ Ori_________________ Languages, meaning they are very easy to write with complex abstractions (i.e. few lines to say something really complex).

2. Being so high-level, they can be used for some really complex stuff, such as simulations, g_________, interacting with images such as maps, and different applications (programs).

3. Objective C and C++ are both based off of the ___ Language.

It should be noted that JavaScript works with objects as well, and Python is also considered (at least partially) an interpreted language. It should be no surprise that "hybrids" like this have become popular in the context of Internet-based programs that are interpreted by browsers, yet can be written at a very high level.

Complete G after you have completed all prior letters.

G. "The Evolution of Computer Languages"

1. Upward arrows show a sort of genealogy of popular programming languages. For instance, you see _____ (count) types of BASIC that have become popular. The most recent version on this paper became popular in the 19___'s. M__________________ has developed V_____________ Basic for tools like Excel and utilizes developments in O_______________O________________ Programming

2. The base of this genealogy is the language F_____________________ which is used in eng____________________ even today.

3. One branch of this genealogy ends with ADA. Why is ADA not so popular?

4. Java in another end of this genealogy. Java is popular because it can run on ____.

Java is derived from C++ which was based on C, which is based on A___________ and F____________.

5. LOGO is still being used today to hook kids into programming, using graphics and lists (it is derived from LIS__). It led to development of S_____________________ that was a huge influencer of modern languages as it uses O____ P___________________________ and used a G______________________ U ___________ I_______________________.

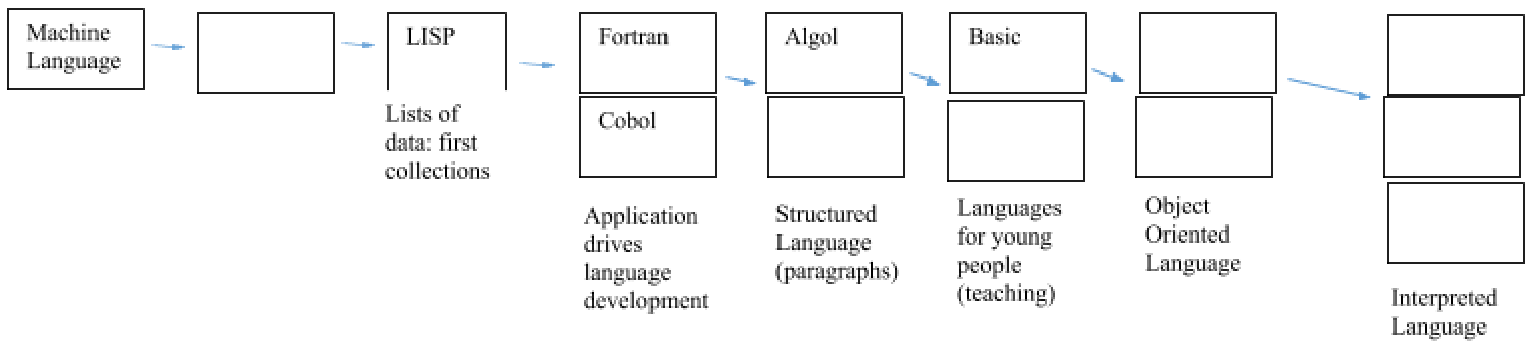
6. APL lost a lot of popularity because programmers needed new hardware to code with it. What was this hardware? A s________________________ k__________________________

7. For the following, place a 'X' to indicate if this feature would improve the popularity of the language or decrease its popularity. The table is on the next page.

Language…	Improves popularity	Decreases popularity
Requires special hardware to make		
Is super concise so it's hard to read		
Uses structures to improve organization – like paragraph blocks, line breaks, page breaks		
Uses logic so complex relationships can be used to make a computer artificially intelligent		
Does not come with error-checking tools		
Uses very little memory		

Fill in the following Language Evolution Chart with the languages: Assembly Language, C, C++, HTML, Java, Javascript, LOGO, PHP.

Low-Level →→→→→→→→→→→→→→→→→→→→→→→→→→→→→→ High-Level


Machine Language
LISP
Lists of data: first collections
Fortran
Cobol
Application drives language development
Algol
Structured Language (paragraphs)
Basic
Languages for young people (teaching)
Object Oriented Language
Interpreted Language

Complete one of the last letters: H, I, or J

H. "Evolution of the Mobile Phone"

1. Mobile phones are reported to be the ________ rapidly adopted tech. in human history.
2. The first commercially available mobile phone was available in ____________. That's just _____________ years ago!
3. List five features that improved the popularity of the phone over its evolution.

I. Evolution of the Video Camera *"destrod" probably should be "destroyed" in this paper, and "launcheched" should be "launched"*

1. The commercially available video camera was first made available in 19____. That's ____ years ago!
2. Loose film was eventually replaced by a plastic-box surrounding film or tape. Why would the plastic-encased film be an improvement?

Think about the further evolution of storage media from 1995 on. How does this improve on the film/plastic-encased film storage?

3. Look at how the different cameras appear to be carried. What are improvements with respect to this, over time? Consider the experience for the person holding/using the camera.

J. The Evolution of the Camera

1. Pinhole cameras became popular around ____00. That's about _____ hundred years ago!
2. Consider the size and precision of the image. How has that changed over the camera's evolution? Use specific examples based on the paper.
3. Consider how the user of the camera holds/uses the camera. How has this been improved across the evolution of the camera? Use specific examples based on the paper.
4. Think about the camera from 1975. It was actually a gateway in terms of image storage. Consider what preceded it and what came later, and explain what was so revolutionary about it.

Group project: *If you have extra time, you can start thinking about your ideal topic…*

Students will explore products (hardware or software with an emphasis on process) that one might expect improvements of, justifying the changes, then querying the designers to understand why a seemingly archaic representation is maintained. This will involve writing questions, getting feedback from parents/an older generation, then sending them to professionals for feedback. Student groups will generate a poster in slide format that will summarize their product's evolution over time, and lessons learned about trade-offs including input from the professional.

Great examples from 2016:

Ayervedic versus Western Medical Treatment

Vinyl versus Digital Audio Media

Real time/space shopping (even with cash) versus Online Shopping