First, <https://xkcd.com/927/>

|  |  |
| --- | --- |
| [https://qph.ec.quoracdn.net/main-qimg-f6730132df8eba09ebb81aff33e5ae11](https://www.explainxkcd.com/wiki/index.php/927:_Standards) | *The great thing  about standards  is there are* so many *to choose from.* |

**Do TWO of the following questions writing 250+ words each.   
Or, do ONE question in 500+ words with a deeper exploration.**Address only *one or two* questions. Please focus your efforts. Many short answers do not allow for thoughtful responses. There are options this week due to the breadth of our topics; you can even make up your own question.

**The notes under the 🡺 main question? are there to stimulate thought and help frame the question; they are not meant to be prescriptive.**

**N.B.   
The value of any course's assignment is not the answer or the marks, it is in your *development* of an answer.**

Please do not copy things from CourseHero or similar sites. Such action attracts Academic Honesty reports for plagiarism.

**Avoid the use of paraphrasing tools.** "Evade the utilization of rephrasing implements" is a sample of what you'll get: the poor use of a thesaurus by a mindless algorithm. Paraphrase generators do not hide plagiarism, they make it glaringly obvious. A submission containing any text that's been through a paraphrase generator may get zero marks overall. Paraphrased answers without attribution, i.e. plagiarism, will certainly be marked as zero. The practice may also result in an Academic Honesty report.

Delete this paragraph and everything above.   
Keep only your identifiers and the 🡺 question(s) you answer in the submitted docx file.

🡺 Your name: Student No.: UserID: *user* @mySeneca.ca

🡺 **When does Hardware matter…to You?**

Computer hardware ranges from mainframes to smartphones to embedded devices within the Internet of Things. What kind of software would you like to write, what language(s) would you like to use? This begs the question, what kind of hardware might it run on? When does hardware matter to you? When doesn’t it? Or, does cloud computing make the question moot?

🡺 **Which computer language is best suited to writing software for which of the many hardware platforms?**

Some hardware has levels of software from device drivers to Operating Systems to programming compilers, utilities & frameworks, to user-level apps. Which computer language(s) might be best suited to writing software for which of the many aspects of hardware…and why?

🡺 **Write a Pseudocode program independent of any computer language.**

All software languages have three kings of logic: sequential, decision (selection/conditional), and iterative (looping). Pseudocode has those same structural conventions but is intended for human reading rather than source code compiling. Pseudocode is a clear set of instructions, a procedure to do something, an algorithm.  
Pseudocode should be able to run on a human with pencil and paper.

Using the three kinds of logic, write the pseudocode instructions for a robot to…

* Select a dress shirt from the closet, put it on, and button it up.  
  *OR*
* Make a sandwich for you  
  hint: Google videos for “algorithm sandwich”

**N.B. How hard can it be?** There is much [tacit knowledge](https://en.wikipedia.org/wiki/Tacit_knowledge) you know *without thinking* that a robot also needs to complete the task without wrecking your home. E.g. you ensure a door is open before getting something out of a storage area.

🡺 **Just how many different programming languages do we really need?**

One answer might be that there are different types of jobs a computer must do and various languages are better suited to solving diverse kinds problems. OK, but how many languages and for what purpose?

On the other hand, as computer hardware gets faster and faster, the performance difference among languages and language types might become insignificant. Could a single language become the one universal all-purpose language?

Make a case in favour of the many and against the one (or the few).  
*OR*  
Make a case against the many and in favour of the one (or the few).

🡺 **Can recipes be considered as programs?**

Is a recipe really an algorithm to solve a problem? Programming uses sequential, iterative, and decision logic to implement an algorithm. Are those three types of logic sufficient to produce the result of a recipe? The application of an algorithm *in various environments* can be the real challenge. Systems people call those Use Cases.

Tea is the most popular drink on the planet. The algorithm/computer/robot must distinguish between making tea, drinking tea, and the phrase "Let's have tea" which usually means both.   
"Hey robot, make tea." — "Tea is made by the Camellia Sinensis plant. I'm a robot."  
"Hey robot, make me a cup of tea." — "It is impossible to make a cup from tea."  
"Hey robot, let's have tea." — "You already have tea. It's in the cupboard. Or in the Roman alphabet between S and U." And stop waking me while I'm recharging."

Assume there is a selection of different kinds of tea available for making tea.   
"Put tea into a pot or cup, add H2O at 100°C until level is 15mm from rim".   
It sounds simple but does all tea require 100°C H2O? How much tea relative to water and what are the units of measure? How long should the tea steep? Should the tea leaves be left in the pot/cup or removed after steeping? How would you make tea in your kitchen, at Seneca, camping in the wilderness during summer or the depths of winter, on the International Space Station? Is there an algorithm to make any kind of tea…anywhere…under any conditions?

🡺 **Make up your own question germane to this week's ideas and answer it  
… but it must be an *interesting* question.**

Examples of questions other students have asked (you can use one of these):

* How have women been important to the development of computing? And what now?
* Why is the history of computers and programming relevant and how can you use it to advance your career/studies?
* Has ICT really revolutionized the world or has it just made it faster and is faster good enough? After the agricultural and industrial revolutions, have we progressed beyond the old, are we in the new information revolution, or are we [past it?](past%20it?)
* What makes C an evergreen language? How and why did it become old enough to be thought of as evergreen?
* Cats...are they ruining the Internet or responsible for its development?
* Are humans misusing the Internet and its ability to spread provocative information like wildfire? Yes, of course they are. Why is that happening? What can we do about it?