

Don't be a cheater.

The value of any assignment is not the answer or the marks, it is in your *development* of an answer.

ChatGPT is permissible when the entire session is added as an appendix to the answers. A few marks are awarded for analytical and probing prompts, nothing for ChatGPT's output, but many marks for your insightful and informed critique on ChatGPT's output.

Generative AI is manufacturing, not creating. Generative AI is associative, not creative. Generative AI constructs prose, not ideas.

If educators believed a web search of questions and copy & paste of answers was scholarship, we would give you the answers with the questions and skip the marking. Professors can submit assignment questions to ChatGPT as easily as you can.

Assignment and quiz 'answers' from other students can be found through a web search. Using academic work found on 'study' (predatory) websites is a violation of copyright and academic integrity. So is posting it there. The 'answers' are usually wrong, and glaringly so. (Good students tend not to post their work for use by others.)

Only journals, library resources, published books, and original source websites (e.g. Microsoft.com) count as research for ideas learned or quotations taken to support your work.

Academic referencing of sources is standard practice for quotations or paraphrased ideas you present in your submission. [Attributing the sources](#) from where we learned something is an ethical tradition: to recognize and honour those from whom we learned those ideas & words, so that we represent ourselves fairly, and to be mindful not to confuse ourselves, or have anyone else confuse us, with the source.

Marks are given only for your original work, not anyone else's.

Facility with English

Marks are awarded for the clear explanation of concepts and ideas. Marks are not deducted for poor spelling, grammar, or usage. However, without proper spelling, good grammar, and appropriate usage, the explanation of concepts and ideas will suffer and so will the marks.

N.B. The use of Grammarly or similar services as an editor to correct grammar, spelling, and polish your writing style is permissible only when all your original work is submitted along with the edited version. Otherwise, your ideas can easily appear to have been plagiarised and/or AI generated.

Submissions are checked for plagiarism and AI content using an industry standard service that is focused on the academic integrity of submissions. It identifies ChatGPT's output as from an AI, and correctly identifies bona fide student answers as created by humans.

Paraphrasing does not count as original work – it barely qualifies as a grammar exercise.

Paraphrasing is plagiarism if the source and the extent of paraphrasing is not accurately cited and referenced. See Seneca's academic integrity resources.

"Evade the utilization of rephrasing implements" is the result from a paraphrasing tool for "Avoid the use of paraphrasing tools." Paraphrase generators do not hide plagiarism, they make it glaringly obvious because their mindless use of a thesaurus results in poor English usage. A submission containing any text that has been through a paraphrase generator will certainly receive fewer marks due to poor readability. Paraphrased answers without attribution – [plagiarism](#) – will result in an assignment mark of zero or less, and an Academic Integrity Report with additional [consequences](#).

Translating

Seneca is quite clear about its expectations of English literacy. If you need a translation tool, you have not met the expectation. (It is okay to use a *language*-to-English dictionary to confirm your use of terms.)

Automated translation tools produce text similar in quality to paraphrasing tools: extraordinarily awkward and mostly unreadable. [Here is an example](#). If it is mostly unreadable, it is mostly unmarkable.

More than just your [opinion](#) is being asked in this activity. An [opinion](#) is subjective, it is your point of view, your preference, what matters to you, and nobody is obligated to listen to it. If you want others to consider your standpoint, a reasoned [argument](#) supported by facts, principles, and analysis is required. An argument carries weight, could change others' point of view, and could matter to many other people. See [No, you're not entitled to your opinion](#).

- ✓ **Note for any course:** answers copied and pasted without citation and referencing will result in a minimum of zero marks **FOR THE ENTIRE SUBMITTED WORK**, may incur a sanction (a negative mark), and may be subject to Academic Integrity review.
- ✓ Please DO *discuss* aspects of the activity with your colleagues and professor but create the answer in your own words according to your own understanding. The integrity rule: **talk all you want, read all you want, but don't copy/paste anything** without attribution, citation, and references. Share ideas, not files. Share understanding the questions, not the answers.
- ✓ "Study resource" sites like [chegg](#), [studocu](#), and [coursehero](#) want to be searchable to attract paying customers, that is, students. File sharing through those sites is not helping each other or quid pro quo. ("Upload your study documents for free access to other study documents in our library.") It is predatory behaviour by commercial third parties.

Select:

- A single logic task from Algorithms and Pseudocode
- A single topic ICT Past, Present, and Future
- Write 250+ words for each.

Note that the minimum word count gets more or less average marks depending on the quality of content. To go above average, see the marking rubric in the course Announcements.

The notes under some questions are there to stimulate thought and help frame the question; they are not meant to be prescriptive.

Algorithms and Pseudocode

Answer **one** of the following logic questions. *In addition, and regardless of whether your pseudocode ran to 250+ words, critically assess how much general knowledge and “common sense” is needed to carry out your algorithm or pseudocode successfully.* An [algorithm](#) is a step-by-step procedure to solve a problem. [Pseudocode](#), i.e. fake code, is generic programming instructions not specific to a language. For our purposes, consider algorithm and pseudocode equivalent. There is no need to discuss what algorithms or pseudocode. You are taking ICT courses, we assume you already know this.

All software languages have three kinds of logic: sequential, conditional (decision | selection), and iterative (looping). Pseudocode includes those structural concepts but is intended for human reading rather than source code compiling, so the only proscribed syntax and style is *clarity*. The most useful format for an [algorithm](#) or [pseudocode](#) is a numbered list, *not prose*.

How hard can it be? There is much [tacit knowledge](#) you know that a robot does not know to complete the task without wrecking your home. E.g. you know to open a cupboard's door before trying to get something in or out of it.

This is a two-part question.

Select **one** of the following:

1a. → **What is the...**

- programming a robot would need to cook something if given a recipe?
- algorithm to decide what we are having for dinner?
- pseudocode for a robotic vacuum to clean your floors?
(without destroying the place or giving the cat PTSD)
- pseudocode for a robot to clean your dishes? (without using a dishwasher)
(You don't get anything clean without getting something else dirty. —Cecil Baxter)
- pseudocode for a robot to empty the dishwasher?
(so as you can find the cutlery and crockery later, please)

1b. → **What general knowledge and “common sense” is assumed in order to carry out *your* algorithm in 1a. successfully?**

- Clarifying assumptions about what the solution depends upon is perhaps more important than the instructions themselves. Computers and robots don't know what you mean, just what you say.
- As an example, [HERE](#) are assumptions programmers make about time that cause problems.

ICT Past, Present, and Future

Select **one** → question ...

→ **Just how many different programming languages do we really need for high level, general purposes, e.g. as used for user-facing application programming?**

N.B. Of course, low-level languages are needed for hardware specific purposes: operating systems, device drivers, compilers, virtual machines, and embedded devices. Those low-level uses of languages are outside the scope of the above question.

As computer hardware gets faster and more capable, the performance difference among the various high-level, general purpose language types might become insignificant. Could a single language become the one universal high-level, all-purpose application 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?

→ **How does cloud computing change our need for personal computing hardware?**

Computer hardware ranges from mainframes and servers to personal computers to smartphones to embedded systems. The Internet brings it all together. Many feel as though marooned on a desert island when a network connection is not available. Assuming effective HCI (human computer interaction) along with reliable and fast-enough communications, does cloud computing make the end-user's hardware irrelevant? Does the cloud move performance issues away from personal hardware? (Do you really, really need the latest smartphone?)

→ Cats...are they ruining the Internet or responsible for its development?

Giving credit where it is due... the following are worthy questions students have posed and answered (you can select from these, too):

→ Has ICT revolutionized the world, or has it just made the same old things faster?

(same old: television, radio, cinema, newspapers, pamphlets, libraries, manuscripts, concert hall, theatre, town square, scrolls, alphabets, cave paintings, people conversing)

→ Are we in a new information revolution?

"There is a theory which states that if ever anyone discovers exactly what the Universe is for and why it is here, it will instantly disappear and be replaced by something even more bizarre and inexplicable. There is another theory which states that this has already happened." – [Douglas Adams](#)

→ How do we address the fear of artificial intelligence in a productive way?**→ Make up your own question germane to this week's ideas and answer it, but it must be an *interesting* question.**

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

The great thing about computing standards is there are so many to choose from.