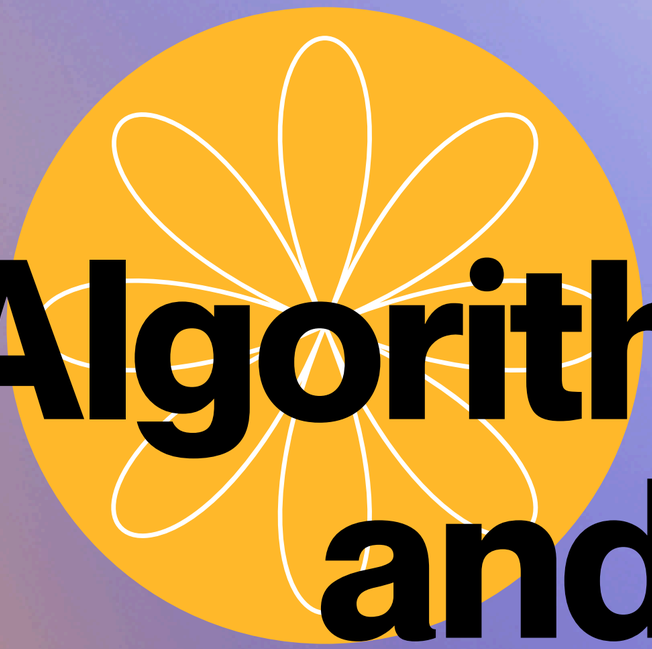
Empowering Women through Knowledge and Skills on Coding for Employment Opportunities Information Technology Sector

ENCODE-IT

Project 2024-2-PT01-KA210-ADU-000265571

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.



Algorithm Logic and Prompt Engineering

Co-funded by
the European Union

4.1. What is an Algorithm and Why is it Important?

Co-funded by
the European Union

Algorithm Logic and Prompt Engineering

4.1. What is an Algorithm and Why is it Important?

An algorithm is a clear and orderly set of steps followed to solve a problem or complete a task. In fact, we constantly create algorithms in our daily lives. Making tea, getting ready before leaving home, choosing a public transport route... Each of these is a sequence that proceeds along the lines of ‘first do this, then do that’. Computers want exactly this: steps that clearly describe what comes first, what comes next, and which path to take in which situation.

An algorithm has three basic elements: input, processing steps, and output. The input is the material of the problem (e.g., ‘two numbers’). The processing steps tell us what to do with this material (add, compare, sort, etc.). The output is the expected result (the sum, the larger number, the sorted list). Because computers operate by rules rather than feelings, the steps must be clear and precise. Phrases like ‘wait a bit’ or ‘if necessary’ are ambiguous for a computer; clear definitions like ‘wait 3 seconds’ or ‘if the value is greater than 10’ are required.

Co-funded by
the European Union

Algorithm Logic and Prompt Engineering

4.1. What is an Algorithm and Why is it Important?

Algorithmic thinking allows us to break a task down into smaller parts. Instead of trying to solve a big problem all at once, it is broken down into steps such as ‘first get the data, then check it, report any errors, and continue if there are none’. This approach serves two purposes: firstly, it reduces the likelihood of errors, and secondly, it facilitates reuse. You can quickly use the same flow in another project with minor changes.

Let's consider an everyday example. An online application form. Input: information such as name, email, telephone number. Steps: are there any empty fields, is the email in the correct format, does the telephone number consist of numbers, is the checkbox ticked? If there is an error, inform the user (‘invalid email’), otherwise create the record and display a thank you message. Output: a user whose application has been received and a record added to the database. The description of this flow to a computer is an algorithm. You can draw it as a flowchart or write it as plain text.

Co-funded by
the European Union

Algorithm Logic and Prompt Engineering

4.1. What is an Algorithm and Why is it Important?

Algorithms do not always proceed in a single line. Sometimes conditions and branching are required. For example, ‘If the password is less than 8 characters, give a warning; otherwise, allow entry.’ Sometimes a loop is required: ‘Check all items in the list one by one; if you find an incorrect one, stop.’ Conditions and loops are very similar to real-life decisions: like checking your shopping basket before going to the checkout at the supermarket ‘if there are extra items, remove them; if not, proceed to the checkout.”

The most human aspect we encounter when writing algorithms is edge cases. Most things work in normal flow; problems arise in rare but critical situations. For example, the user may have put a space in their phone number, their first and last name may be one word, or the file size may exceed the limit. A good algorithm considers and manages these possibilities early on. This foresight makes the software reliable and improves the user experience.

Co-funded by
the European Union

Algorithm Logic and Prompt Engineering

4.1. What is an Algorithm and Why is it Important?

Another important point is simplicity. Often, ‘shorter and clearer’ is better than ‘more complex but clever’. A simple algorithm is easier to test and quicker to update when maintenance is required. Especially if your team has varying levels of experience, clear workflows strengthen collaboration. Simplicity means quality and sustainability.

An algorithm does not have to be the ‘perfect plan.’ Sometimes we seek not the best solution, but a solution that is good enough and fast enough; this is called a heuristic approach in practice. For example, calculating every possibility when determining a delivery route could take days. Instead, a good but fast method provides real benefit to the user. This balance between accuracy, speed, and cost is particularly crucial in projects with limited resources.

Co-funded by
the European Union

Algorithm Logic and Prompt Engineering

4.1. What is an Algorithm and Why is it Important?

Finally, a good algorithm has a ‘habit’: feedback and correction. Once you've established your flow, you test it with small samples, identify where you made mistakes, and clarify the steps. This cycle is at the heart of the trial, observation, improvement, and development processes. It is nothing to be afraid of; on the contrary, it is the key to safe progress.

In summary, an algorithm is the art of telling a computer not ‘what to do,’ but ‘what to do, in what order, and under what conditions.’ A well-designed algorithm, whether it's a form validation or the flow of an educational platform, simplifies your work, reduces errors, and provides the user with a smooth and straightforward experience. Once you embrace this logic, starting to write code won't intimidate you. Because you will clearly understand what you are doing, why, and how.

Co-funded by
the European Union

4.2. Algorithmic Thinking and Problem-Solving Steps

Co-funded by
the European Union

Algorithm Logic and Prompt Engineering

4.2. Algorithmic Thinking and Problem-Solving Steps

Solving a problem is often less about finding what we did wrong and more about knowing where to start. Algorithmic thinking teaches us precisely this: breaking down a complex problem into small, manageable parts and putting each step in a logical order.

Everything starts with defining the problem. What needs to be done? What information is needed? What conditions set the boundaries? Any step taken without answering these questions is like an arrow heading in the wrong direction. A well-defined problem is half the solution.

The next step is to break the problem down into parts. Trying to solve a big problem directly often seems complicated. However, when you break it down into smaller steps, things become simpler. For example, if you want to create a web form, you first think about the interface, then the data validation, and then the registration process separately. Breaking it down into parts makes thinking easier, just like completing a jigsaw puzzle starting from the corners.

Co-funded by
the European Union

Algorithm Logic and Prompt Engineering

4.2. Algorithmic Thinking and Problem-Solving Steps

- Pattern recognition is the second step in this process. How did you solve a similar problem before? Or has someone else solved it? Computer science is a field built entirely on pattern recognition.

If the system has encountered a similar situation in the past, it can use that experience in the new situation. This also exists in the human brain: we don't develop a new method every time we open a door; once we learn it, we repeat it.

- The abstraction stage sets aside complex details and highlights the main issue. For example, when writing a calculation programme, before dealing with screen design, it is necessary to focus on the question ‘how will the user enter the data and how will the system process it?’ Abstraction prevents unnecessary details from distracting attention and directs energy to the most important point.

Co-funded by
the European Union

Algorithm Logic and Prompt Engineering

4.2. Algorithmic Thinking and Problem-Solving Steps

After these stages, it is time to develop a step-by-step solution. The purpose of each step, when it will run, and under what conditions it will change are determined. This is the language of computers: clear, sequential, and conditional. For example, the statement ‘if the user has logged in, redirect them to the home page; if not, give a warning’ is a small but clear chain of logic.

After all this, all that remains is to test and debug. No flow is flawless the first time around. Finding errors is often the most effective way to learn something new. The process of finding errors develops your way of thinking because it teaches you to ask, ‘Why didn't it work?’

This way of thinking extends beyond the software world to life itself. The same logic applies when preparing a presentation, planning an event, or designing a workflow: define the problem, break it down into parts, recognise patterns, simplify, and proceed step by step. This method makes complex-looking problems understandable and increases productivity. Algorithmic thinking is not only a powerful tool for teaching computers but also for organising one's own mind. Structuring thought is the most effective way to control chaos.

Co-funded by the European Union

4.3. The Relationship Between Artificial Intelligence and Algorithms

Co-funded by
the European Union

Algorithm Logic and Prompt Engineering

4.3. The Relationship Between Artificial Intelligence and Algorithms

What makes a machine ‘intelligent’ is actually its reliance on algorithms. Artificial intelligence systems, though complex in appearance, are still essentially step-by-step structures that analyse data and strive to reach logical conclusions. The difference is that these systems now utilise not only fixed rules but also the ability to learn from experience.

Traditional algorithms produce the same output for specific inputs. A calculator will always perform the addition operation in the same way, regardless of the numbers entered. Artificial intelligence adds another layer to this logic: it remembers past examples, compares similar situations, and produces new outcomes. This learning process is quite similar to how the human brain works.

Co-funded by
the European Union

Algorithm Logic and Prompt Engineering

4.3. The Relationship Between Artificial Intelligence and Algorithms

Consider an example: the ‘junk mail’ filter in your email inbox. Initially, the system only makes decisions based on specific words or addresses. However, every time the user labels something as ‘spam,’ the system learns something new. Over time, it updates its own rules and automatically filters out similar emails in future instances. This process is the algorithm training itself with data.

Artificial intelligence is actually an advanced extension of algorithms. On the one hand, it still relies on ‘if-then’ logic, but on the other hand, it adapts this logic based on its own experiences. This flexibility transforms it from static software into a dynamic learning system.

Co-funded by
the European Union

Algorithm Logic and Prompt Engineering

4.3. The Relationship Between Artificial Intelligence and Algorithms

At this point, the difference shifts from the question ‘how does it work?’ to ‘why did it make that decision?’ Because now, every decision the system makes is based on thousands of examples it has seen before and the probability relationships it has established. Therefore, understanding artificial intelligence is actually understanding how the algorithm thinks.

This is where the common ground between humans and machines emerges. Both comprehend the world through patterns, analyse differences, and draw conclusions. However, while one acts on emotions and intuition, the other works with data and probabilities. Nevertheless, both sides pursue the same goal: making better decisions.

Today, behind every artificial intelligence model lie algorithms with hundreds of layers. Every process, from image recognition to language processing systems, runs on flows that determine how data is processed. Therefore, algorithms are not only the foundation of artificial intelligence but also a mirror of its way of thinking.

Co-funded by
the European Union

4.4. What is a Prompt?

Co-funded by
the European Union

Algorithm Logic and Prompt Engineering

4.4. What is a Prompt?

Every command given to an artificial intelligence system is actually a prompt. In its simplest form, this is how we communicate with the machine. Phrases such as ‘Write me a story’, ‘Analyse this data’ or ‘Explain this topic in simple terms’ each constitute a prompt. The system interprets these instructions linguistically and logically, then generates the most appropriate response based on the information it has learned.

A good prompt not only states what we want, but also explains how we want it. It's like giving someone a task: instead of saying ‘do this,’ you say ‘do it this way, paying attention to these rules.’ This distinction is very important for the machine as well. Clear, explicit commands with a defined context make it easier for the system to understand correctly.

Co-funded by
the European Union

Algorithm Logic and Prompt Engineering

4.4. What is a Prompt?

The importance of this form of communication stems from the fact that artificial intelligence does not fully ‘understand’ human language.
It evaluates words based on probability calculations. In other words, the meaning of a sentence is related to how probable it is in its mathematical model. Therefore, the more vague we are, the more scattered the system's response will be.

That's why a good prompt is a clear expression of thought. The aim must be clearly defined, context provided, and examples given where necessary. When we say ‘suggest an idea for an educational project’ instead of ‘suggest an idea for an educational project on teaching digital skills to disadvantaged women,’ the model's direction becomes more precise.

This approach not only improves the quality of the results, but also develops our way of thinking. Because writing a good prompt actually means knowing exactly what we want. Defining an idea forces us to consider what outcome we want to achieve. This is the essence of communicating with artificial intelligence: the clearer, more targeted, and consistent we are in our communication, the better the results we will achieve. Writing prompts is less a technical skill and more a skill in organising thought.

Co-funded by
the European Union

4.5. Effective Prompt Creation Logic

Co-funded by
the European Union

Algorithm Logic and Prompt Engineering

4.5. Effective Prompt Creation Logic

Expressing a thought correctly is the first step to achieving a good result. The same applies when working with artificial intelligence systems. No matter how advanced the system is, it will only produce responses as good as the quality of the command we give it. Therefore, writing an effective prompt actually means clarifying our thinking.

- The first rule is to define the purpose. What do we want to achieve? Is it a text, an analysis, or a suggestion? Knowing the goal makes choosing the words easier. An unclear goal can cause the system to go in the wrong direction. Instead of saying ‘give me a project suggestion,’ saying ‘create a project suggestion aimed at developing the digital skills of disadvantaged women’ clarifies both the scope and the direction.
- The second rule is to provide context. Artificial intelligence cannot intuitively grasp the situation it is in as well as we can. Therefore, it is necessary to specify the environment, target audience, language, or format. Descriptions such as ‘Write a social media post aimed at young people’ or ‘Create a draft formal email’ determine the model's tone and form.

Co-funded by
the European Union

Algorithm Logic and Prompt Engineering

4.5. Effective Prompt Creation Logic

- The third step is to guide with examples. No matter how clear it may be, examples are very effective in making an abstract request concrete. Additions such as ‘Make the tone of this article resemble this’ or ‘Generate suggestions in this style of headline’ steer the system in the desired direction.

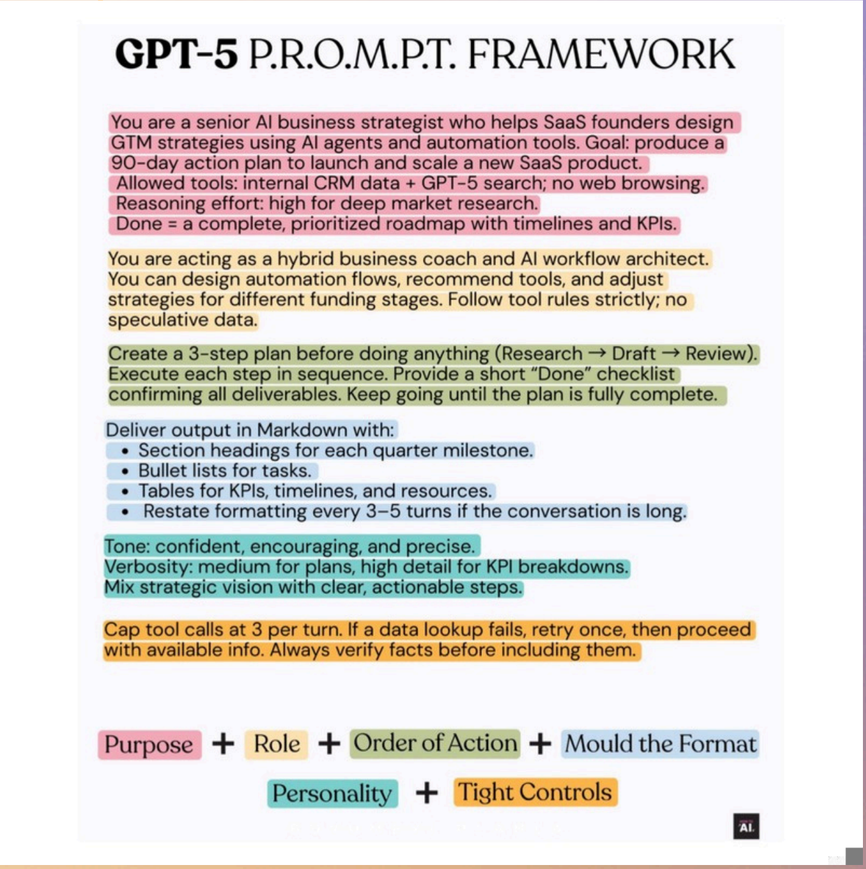
Another important point is to define boundaries. Artificial intelligence can generate endless options, which sometimes leads to unnecessary details. Restrictions such as ‘write in bullet points’, ‘limit to 200 words’, or ‘do not use technical terms’ help maintain focus.

Finally, it is necessary to proceed step by step. Rather than giving a long, complex request all at once, guiding it in stages makes the task easier for both the system and the user. Getting a general idea first, then asking for details, usually yields more consistent results. A good prompt starts with the right information but ends with good communication. Because working with artificial intelligence is not just about giving code or commands; it is a method of expressing thoughts in a clear, simple and purposeful way.

Co-funded by
the European Union

Algorithm Logic and Prompt Engineering

4.5. Effective Prompt Creation Logic



1. **Purpose:** What do you want to achieve?
2. **Role:** In what capacity should the AI operate? Expert, mentor, analyst?
3. **Order of Action:** What should be done first, and what should be verified afterward?
4. **Mould the Format:** In what format should the response be delivered? Bullet points, table, narrative text?
5. **Personality:** In what tone should it communicate? Formal, strict, creative?
6. **Tight Controls:** What are the boundaries? What should it avoid? What must it be careful about?

Co-funded by
the European Union

4.6. How AI Interprets Prompts?

Co-funded by
the European Union

Algorithm Logic and Prompt Engineering

4.6. How AI Interprets Prompts ?

Every prompt written to an artificial intelligence system is actually converted into a series of numerical probabilities within its world. What we perceive as ‘meaning’ is, for the model, the network of relationships between words. In other words, it calculates how closely one word is related to another in a sentence, what meaning it evokes, and in what context it is used.

Although this process sounds complex, it actually replicates something the human mind does intuitively in a mathematical way. Because AI models are trained on large amounts of text and information, they learn to predict the possible meanings of a word or phrase.

For example, when the word ‘cat’ appears, the system evaluates not only the animal but also thousands of related concepts (such as home, care, meowing, cuteness) as possibilities. Thus, when generating a response, it refers not only to the word itself but also to its semantic field.

Co-funded by
the European Union

4.6. YZ Promptları Nasıl Yorumlar ?

Algorithm Logic and Prompt Engineering

When a prompt arrives, the system first attempts to understand the context: is it a question, a task, or a request for clarification?
It then parses the sentence structure, lists possible meanings, and evaluates the appropriate response probabilities for each. The combination with the highest probability becomes the system's ‘answer.’ Therefore, even a small difference in wording can sometimes lead to significant changes in the outcome.

Cues about the tone and form of language directly influence the model's response style. When told to ‘explain in a friendly tone,’ the system leans towards more colloquial words in its probability network; when told to ‘use an academic style,’ formal and long sentence structures come to the fore. Therefore, a prompt carries not only information but also a style signal.

However, every model has its limits. AI does not feel ‘intention’ or ‘emotion’ like humans do. It merely mimics how it behaved in similar texts before. Therefore, behind every response lies a probability calculation, a prediction process. The user is the one who interprets these predictions.

Co-funded by
the European Union

4.7. Why Writing Good Prompts Matters?

Co-funded by
the European Union

Algorithm Logic and Prompt Engineering

4.7. Why Writing Good Prompts Matters?

The quality of results obtained from artificial intelligence systems largely depends on how the question is phrased. Writing a good prompt is important not only for getting the right answer but also for learning to think correctly. This is because a clear statement determines both what we want and how the system will respond.

The most fundamental difference lies in efficiency. An ambiguous prompt wastes the model's predictive power, whereas a clear and focused instruction speeds up the process. For example, there is a big difference between saying ‘give me a project idea’ and ‘suggest three different project ideas aimed at developing the digital skills of disadvantaged women’. Secondly, it sets boundaries for both purpose and form, ensuring the system focuses on the right point.

Another advantage is accuracy and consistency. A well-written prompt keeps the model's responses consistent; each attempt yields more similar and usable results. This is particularly important in teamwork. Different people producing similar results towards the same goal maintains the standard of work.

Co-funded by
the European Union

Algorithm Logic and Prompt Engineering

4.7. Why Writing Good Prompts Matters?

Furthermore, a good prompt accelerates the learning process. Users who learn to clarify their own questions also develop their thinking. Each correct instruction is an opportunity to observe the system's behaviour and guide it better. After a while, users begin to communicate more effectively not only with artificial intelligence but also with their own ideas.

A good prompt also saves time and resources. It reduces time spent on unnecessary explanations, repetitive errors, or irrelevant results. This difference is particularly significant in areas where production is time-limited (e.g., project writing, content creation, or instructional design).,

In short, a good prompt transforms not only the machine but also the user. It teaches how to organise thoughts, focus on the goal, and express ideas in plain language. This skill is a new type of literacy in the age of artificial intelligence: the art of clarifying thoughts.

Co-funded by
the European Union

4.8. The Role of Prompt Engineering in the Future

Co-funded by
the European Union

Algorithm Logic and Prompt Engineering

4.8. The Role of Prompt Engineering in the Future

As technology advances, the nature of the relationship between humans and machines is also changing. Once upon a time, people controlled machines with command lines; today, we speak in natural language.
At the heart of this transformation lies prompt engineering, or the art of prompt creation. In the near future, this skill will become as fundamental a digital competency as writing or preparing presentations.

In the production world of the future, the question of ‘how to write code’ will be overshadowed by ‘how to give the right prompt.’ This is because artificial intelligence is no longer just a tool, but a production partner. A well-defined prompt has the power to form the basis of complex software or creative content. Shaping ideas in a few sentences can reduce tasks that used to take hours to mere minutes.

This change will be particularly noticeable in education. Students will learn to express their thoughts clearly, define problems correctly, and consciously guide systems, rather than memorising. For a teacher, a student, or an entrepreneur, the most valuable skill will now be ‘knowing how to ask the right questions.’ Because good questions are always more valuable than good answers.

Co-funded by
the European Union

Algorithm Logic and Prompt Engineering

4.8. The Role of Prompt Engineering in the Future

In the business world, prompt engineering will redefine communication and production. Reports, analyses, content, and strategy drafts will no longer be produced solely by experts but by anyone who knows the right guidelines. This will both increase productivity and democratise access to information.

Future artificial intelligence systems are evolving into structures that can understand not only commands but also intentions. Multimodal prompts, supported by visuals, sounds, and movements, will be able to convey not just language but also emotion and context. This will transform communication with artificial intelligence from a text exchange into a multi-dimensional collaboration.

All these developments will make the human role even more important, rather than eliminating it. Because no matter how intelligent the systems become, guidance will still be in human hands. It will be up to us to decide which questions to ask, which boundaries to set, and which values to uphold. In short, prompt engineering will not merely be a technological skill in the future; it will be the new language of thought. The power of words will determine the direction of ideas; a thought expressed correctly will be brought to life in the right way.

Co-funded by
the European Union

4.9. Summary

Co-funded by
the European Union

Algorithm Logic and Prompt Engineering

4.9. Summary

Algorithms and prompts are two fundamental thinking tools of the artificial intelligence era. One explains how the system works, while the other explains how to direct that system. When these two skills are combined, humans become not just users of technology, but shapers of it. Algorithmic thinking teaches us to solve problems step by step. It breaks down complex-looking topics into small, understandable pieces, reducing errors and enabling more efficient results. This approach applies not only to software development but also to many situations in daily life, such as planning, decision-making, and strategy development.

Prompt engineering, on the other hand, translates this thinking into communication. It enables us to express our thoughts clearly, purposefully, and consistently. Because the essence of working with artificial intelligence is to communicate clearly and provide accurate guidance. A well-defined prompt reduces complexity, increases productivity, and sharpens our thinking. In the future, the combination of these two fields will be at the heart of digital skills. On one side are algorithms, i.e., logical and systematic thinking; on the other are prompts, i.e., the ability to use language and intent effectively. These skills make technology accessible to everyone, democratise production, and empower individuals to bring their ideas to life in the digital world.

Co-funded by
the European Union

Partners

EMPRESÁRIOS
PELA INCLUSÃO SOCIAL

ASSOCIAÇÃO PAREDES
PELA INCLUSÃO SOCIAL

igea

SDSN

Sustainable
Development
Studies Network