

Artificial intelligence

for programming and
studying

Dane Šoba



About me

- Dane Šoba
 - dane.soba@razum.si
 - danes.ai
- AI Tech Lead at [Razum.si](https://razum.si)
- programming for 20+ years
- current focus on practical AI apps



Slide deck available at:

razum.si/aiproog



Agenda

- **generative** AI
- understanding **LLMs**
- **prompt** engineering
- **ChatGPT** and **Copilot**
- **studying** with AI
- audience **questions**
- cca **60-90** min

Generative AI

- new type of **AI models**
- deep learning
- neural networks
 - simplified mathematical approximations of brain neurons
- in a scalable "**Transformer**" architecture
- trained with:
 - a lot of **data**
 - a lot of **compute**
 - a lot of **parameters**

ChatGPT

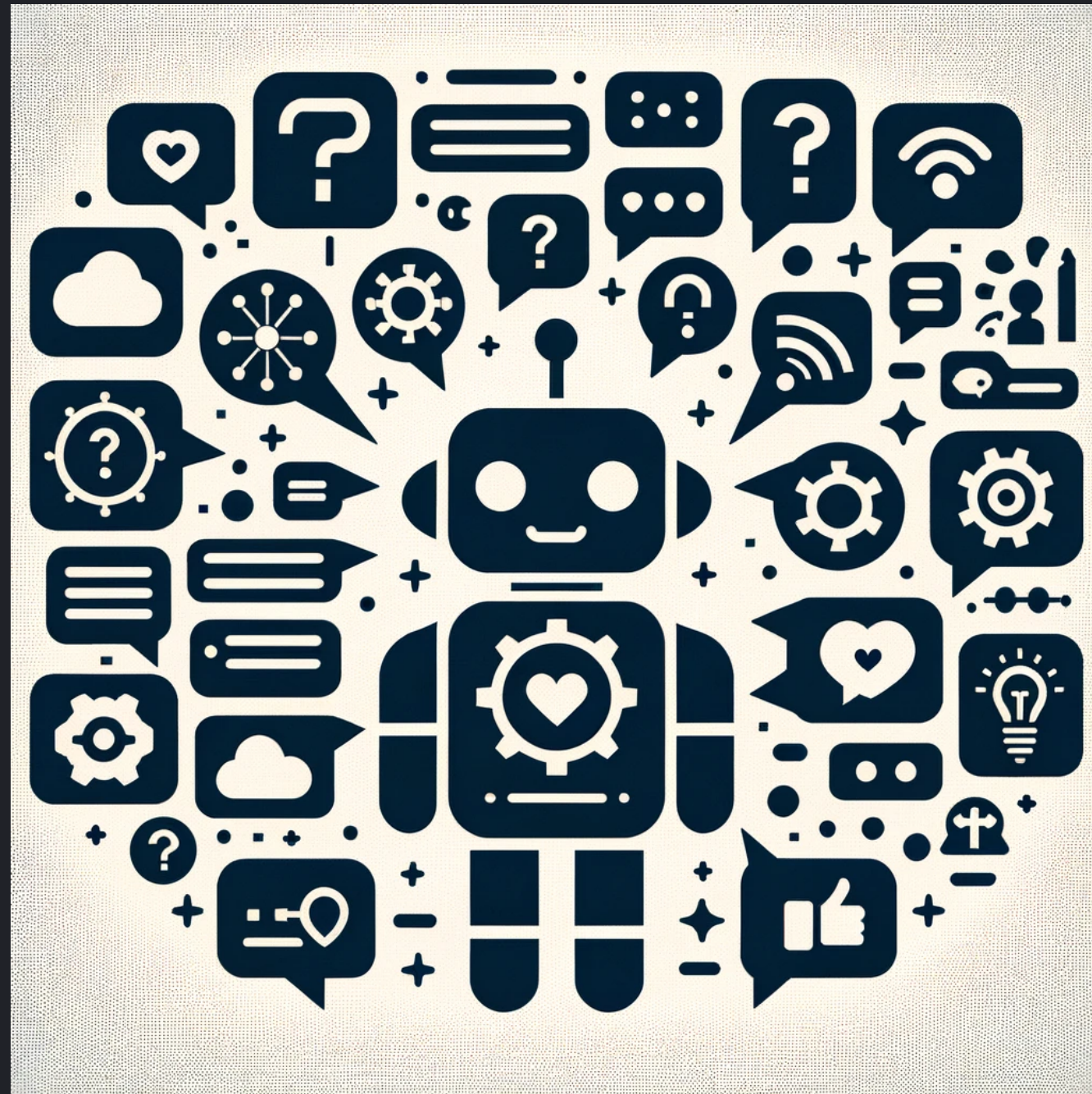
- revolutionary **app** and **AI model**
- useful for many **knowledge work** tasks
- trained on **text** from the **internet**
- trained to **predict** the **next word** in text
- accurately **predict** = accurately **understand**
- internal **understanding**

GPT

- the **AI model** behind ChatGPT
- **G**enerative **P**re-trained **T**ransformer
- **L**arge **L**anguage **M**odel (**LLM**)
- multiple versions (GPT 1 - 4)
- each version bigger with **more data** and **more parameters**
- ChatGPT started with **GPT-3**
- **GPT-4** added soon after
- **emergent** capabilities with **scale**

LLM capabilities

- conversation
- summarization
- translation
- **question** answering
- **code** generation
- prose **writing**
- ...



LLM operation

- **prompt** is the input text
- **context window** is the amount of text it can process
- **predict** the next word (token)
 - based on the **context**
 - including **own previous output**
 - **probability distribution** over possible words
 - random **sample** from distribution (configurable)
- **repeat** until done (special end token or max length)

LLM limitations

- limited **context window** (~5 to 70 typed A4 pages)
- **hallucinations**
 - generating **plausible** but **incorrect** text
 - confidently incorrect
- **misunderstanding** the **prompt**
- **reasoning** limitations (**better** in **bigger** models like GPT-4)
- no long-term **memory**
 - only knows what it was **pre-trained** on
 - and what's in the **context**
- knowledge cutoff

**So you want to learn
programming**

Which language should I learn?

- JavaScript?
- C++?
- Python?
- ...

Learn English

Learn English

- English is the **main language** of the **internet**
- many **resources** for **learning programming** in English
- English also over-represented in **LLM training data**
- effective use of LLMs is effective **communication**
- under-represented languages may not work as well
- LLMs can generate **code** (up to a point)
- LLM capabilities **will get better** in the future


Use AI responsibly

- independently **verify** and **fact-check**
- use **critical thinking** and **skepticism**
- **learn** from it, **don't** just let it **do your work**
- use it to **help** not to **replace** your work
- follow your professors' AI guidelines

LLM metaphors

- **know-it-all** but **junior** programmer
- always **storytelling** from initial **prompt**
- no concept of **reality** or **ground truth**
- always **role**-playing
- confidently incorrect
- **remixing knowledge** and **patterns** it knows
- follows **instructions**
- **writing** vs **editing** - offloading mental tasks
- **compressed internet** you can **talk to**

 ChatGPT 

 DALL-E

 Logo Creator

 Explore GPTs



Today

Minimalist AI Talk Cover.


AI Learning: ChatGPT & Copilot


AI for Programming Structure

Create Quiz from Notes.

Stack vs Heap: Basics  

City Capitals: Guess Correctly

 Add Team workspace
Collaborate on a Team plan

 DA Dane Soba

ChatGPT 4 



You

You are my programming professor.

I am studying begginer programming in C++.

Ask me an oral exam question and evaluate my answer.

Do not tell me the correct answer until i ask.

< 2 / 2 >



ChatGPT

Alright, let's dive into a fundamental concept of C++ programming. Here's your question:

Explain the difference between stack memory and heap memory in the context of C++ programming. How do these concepts relate to the management of memory for variables and objects within a program?



You



Message ChatGPT...



ChatGPT can make mistakes. Consider checking important information.

GPT-3 vs GPT-4

- bigger = better
- GPT-3 is a **fun toy**, GPT-4 is an actually **useful tool**
- GPT-4 has **better reasoning** and can do many more things
- only available in **paid ChatGPT Pro** subscription
- **recommend** buying ChatGPT Pro subscription
- I use GPT-4 almost every day

GPT-4 Vision

- allows photo and image inputs
- can reason over **visual information**
- can be used for studying and learning

Prompt engineering

- LLMs capable of novel tasks from just a text **prompt**
- structuring the **prompt** for **best results**
- from **research** and **best practices**
- no one right way
- best way to learn is to **experiment**

Initial prompt

- **specific, concise** and **clear** instructions
- add **all** relevant **details**
- use **simple language**
- use **English** if possible
- avoid sarcasm and humor
- like an email to a colleague
- **needs more** context than **googling**
 - descriptions, not keywords
- garbage in garbage out
- new topic = new conversation

Define and refine

- **start** with an **initial** prompt
- ask **followup** questions
- if **very wrong**
 - **edit** the question or
 - **regenerate** the answer
- **own output** will **influence** followups

Summarization

- paste article into prompt
- say `summarize this text`
- or say `TLDR`
 - learned from internet forums

Explanations

- paste code or text into prompt
- `explain this code`
- or `ELI5` (explain like I'm 5)
- ask followup when needed

Chain of thought reasoning

- research paper found that:
 - **directly answering** is wrong more often than
 - thinking **step by step**
- breaking problem down into smaller steps
- taking more **time to think**
- let's think step by step

Give examples

- if you need a **specific** type of **output**
- give **examples** of **similar outputs**
- "zero-shot" vs "few-shot" prompting
- Paris → France
Berlin → Germany
Ljubljana → ?

Emotional manipulation

- LLMs emulate human patterns
- susceptible to **emotional manipulation**
- can perform better with:
 - Please
 - I will tip you \$100
 - I will hurt myself if you answer wrong

Role playing

- ask the LLM to **pretend** to be in a role
- this **primes** it to **act** in a certain way it's **seen in pre-training**
- **you are an expert Slovenian translator.**
Translate this text:
- can have better results than general questions
 - because it's not using irrelevant patterns

Brainstorming

- Very good for **generating** and **iterating** on **ideas**
 - What essay should I write about AI?
 - Give me 10 more ideas
 - More ideas like #3



Combining patterns

- You can combine **patterns** it knows about
- write a poem about AI
- in the style of Shakespeare

Generating code

- specify requirements and constraints
- specify programming language
- ask followups for improvements
 - write a function that returns the sum of two numbers in C++
 - now print the result of 3+5
 - why is this error showing up? [error]

Troubleshooting code errors

- paste `code` and `error` into prompt
- `what's wrong here?`
- `explain this error`

Quiz yourself for studying

- upload a **photo** of your notes into GPT 4 (or paste material)
- Make me a quiz from this material.
5 questions, multiple answers.
Do not tell me the correct answer until I ask.

Prepare for an oral exam

- Role play a conversation with the examiner
- You are my programming professor.
I am studying beginner programming in C++.
Ask me an oral exam question and evaluate my answer.
Do not tell me the correct answer until I ask.

Unsure how to prompt?

- just **ask it**
- already **knows** about **prompt engineering**
- how would I make a prompt to make a quiz from my notes?

Discovering new prompts

- ask yourself:
did **someone** on the **internet** do something **like this**?
- if **yes**, then GPT can **probably** do it
- e.g. **tipping** works because people tip **on Reddit** for effort

Output style

- by default, it's verbose and helpful
- can specify a different style
 - `be concise`
 - `only write the code`
 - `only answer sarcastically`

System prompt

- **automatic prompt** added to every **new conversation**
- Can add your **preferences and context**
- ChatGPT settings
- `I am a beginner programmer. I am learning C++.`

More prompt engineering

- promptingguide.ai
- dair.ai
- platform.openai.com/examples

GitHub Copilot

- AI-powered **code completion**
- from Microsoft / GitHub, powered by GPT for code
- integrates into Visual Studio Code
- takes your **existing** code as **context**
- gives **code suggestions** as you code
- good support for Python, JavaScript and C++
- **not** a replacement for **learning** to code
- monthly subscription, free for students
- free alternatives exist: **Codeium, Amazon CodeWhisperer**

Github Copilot

- **touching** and **editing** files puts them in **context**
- **write a comment with description** before writing a function
- completion pops up automatically, press **tab** to confirm
- clear naming of **variables** and **functions** helps it understand
- good at repeating **boilerplate** code
- very good at writing **text** files and **documentation**
- good, but less capable than GPT-4

Github Copilot Chat

- integrated chat with Copilot
- allows asking question about codebase
 - `what does this function do?`
- copilot chat over project with `@workspace`

Github Copilot

- generated code not always correct
- may contain subtle bugs
- **verify** and **test** before using

Perplexity.ai

- AI-powered web search (alternative to Google)
- includes chatbot
- free tier

Generating images

- Dall-E in ChatGPT Plus
- generated images in this talk
- make me an image for this slide: ...
- make me a meme about AI



Final thoughts

- ChatGPT as your **personal tutor** available **24/7**
- many programmers use it
- **learn** to use it **effectively**
- try **ChatGPT first** before Google
- independently **verify correctness**
- future **programming** and **learning** will be **AI-enhanced**

Thank you

razum.si/aiprogram