**Day 1:**

While coding the simple chatbot from the OpenAI API tutorial, I came across some errors in the AI’s responses where it would hallucinate and mix up values from its answers.

A screen shot of a computer

AI-generated content may be incorrect.

At first it would give the correct answer, however when prompted to answer the same question again, it responded with this:

A computer screen with text and numbers

AI-generated content may be incorrect.

I believe that the AI model has taken the value in miles from the first answer, and used it as the value in kilometres for the second answer, showing how it can easily misconstrue the data.

**What is agentic AI:**

* Agentic workflow = iterative and makes the AI do some more thinking/research than usual
* Zero-shot prompting = not asking the AI for any revisions
* GPT-3.5 with an agentic workflow beats GPT-4 with zero-shot prompting

**Agentic Reasoning Design Patterns:**

1. **Reflection:**

Prompt the AI to solve a problem, feed the AI the solution to the problem and ask to fix any errors that emerge from unit tests

1. **Tool use:**

Using the AI to use external tools for research, analysis, and personal productivity

1. **Planning:**

The AI agent breaks the problem down into smaller problems, then plans multiple solutions to solve each problem

1. **Multi-agent collaboration:**

You can prompt the AI agents to have specific roles in a system to solve problems by using specialisation

**What are AI agents?**

* Shift in models from monolithic to compound systems
* Integrate the model into pre-existing processes
* Compound systems are based on the fact that some problems are better if the principles of system design are applied
* Systems are modular
* Compound systems are much faster and easier to adapt
* Compound systems have programmatic control logic (human, think fast) or agents (LLM, think slow)
* Agentic approach is when the agent takes control of the logic
* LLM agents have the ability to reason, act (via tools) and access memory
* ReAct agents – user query -> plan/think -> act (tools) -> observe -> answer
* Narrow problems -> programmatic route
* Complex problems -> agentic route

**Day 2:**

**Prompt Engineering:**

* Prompts can have: input/context, instructions, questions, examples and output format
* Use cases: summarisation, classification, translation, text generation/completion, question/answering, coaching and image generation
* Tips:
  + Use direct instructions
  + Provide any relevant information or data as context
  + Give examples
  + Specify the output format
  + Encourage the model to be factual
  + Align prompt instructions with end goal of task
  + Use personas to get more specific voices
* Controlling output using prompting techniques:
  + Length control
  + Tone control
  + Style control
  + Audience control
  + Scenario-based guiding
  + Chain of thought prompting
  + Avoid hallucinations

**Prompt Engineering pt2:**

* Add the task to the prompt, then the context and audience, then the persona, then the format and finally some training examples -> few-shot prompting
* Define the output format clearly
* Use instructions instead of constraints
* Build a personal prompt library
* Step-back prompting -> target the problem first, then the AI can come up with a more innovative result
* Create AI agents for specific roles
* Meta prompting asking the AI to give you prompts

**Day 3:**

**Web Scraping:**

* Use beautiful soup 4 library paired with requests library
* Look at the html source code of the website and find the tags/classes you want to extract
* Pass them into beautiful soup 4 and use the requests library to retrieve the information
* Format the data in an easy-to-understand way and store the data in files for reference
* The data can then be analysed or used in many ways

Initially came up against many different errors and was referencing the wrong section of the website to be scraped. After lots of trial and error, I realised there was I was referring to the wrong element, so I changed from referencing a class to referencing the relevant tag as well as the id. Using the web scraping tools I was given, I created a program which would retrieve all the undergrad degrees that are able to be studied at Glasgow university, then would prompt the user to input the degree they wish to do. Then, the program would go to the specific webpage for that degree, then tell the user what the grade and subject requirements are for that specific degree. I have a video of the code working attached to the email.

**Day 4:**

**Reading and writing to files:**

* Open file using to either read or write or both using r+/w+
* Make sure to always close the file after editing or reading

**JSON library:**

* Can convert json data to python
* Can be brought into the program using “json.load()”
* Can be deleted using “del”
* Can add data using “json.dump()”
* Can be indented by passing in the indent = n statement into the dump function
* Json can also be used to create dictionaries, so data can be stored in a way where specific data is readily available

**Day 5:**

**To-do List:**

* A python list that can store tasks that need to be done
* The function can create, remove or print tasks that need to be done
* Will be called by the AI when the user needs to access the list
* Can keep track of student homework that needs to be done

**Calculator:**

* A simple calculator created using an iterative function
* The AI will also call this function whenever the user needs to have quick access to a calculator to aid with homework

**AI Model:**

* Will store data such as the to-do list or results from maths equations in txt files
* Will call functions necessary to keep the user satisfied, two of which are the to-do list and the maths functions

**Day 6:**

**To-do List:**

* A python list that can store tasks that need to be done
* The function can create, remove or print tasks that need to be done
* Will be called by the AI when the user needs to access the list
* Can keep track of student homework that needs to be done

**Calculator:**

* A simple calculator created using an iterative function
* The AI will also call this function whenever the user needs to have quick access to a calculator to aid with homework

**AI Model:**

* Will store data such as the to-do list or results from maths equations in txt files
* Will call functions necessary to keep the user satisfied, two of which are the to-do list and the maths functions