

Worksheet: LangChain Orchestration & LCEL

This worksheet guides you through implementing an automated contextual text generation pipeline using LangChain Expression Language (LCEL).

Step 1: Set Up Dependencies

```
pip install langchain langchain-openai
```

Step 2: Initialize Components

```
from langchain_openai import ChatOpenAI
from langchain_core.prompts import ChatPromptTemplate
from langchain_core.output_parsers import StrOutputParser

prompt = ChatPromptTemplate.from_template("Generate a
professional summary for: {input}")
llm = ChatOpenAI(model="gpt-4o")
parser = StrOutputParser()
```

Step 3: Assemble the Chain

```
chain = prompt | llm | parser
```

Step 4: Execute the Chain

```
result = chain.invoke({"input": "Module 9 Learning
Objectives"})
```

Exercise: Automate in a Data Script

Complete the following task: Iterate through a list of dictionary objects and generate summaries for each input.

Tasks:

1. Create a list of 3 topics.
2. Write a loop to invoke the chain for each topic.
3. Print the resulting strings to the console.

Sample Answer

```
topics = ["LCEL Basics", "Prompt Chaining", "Output  
Parsers"]  
  
for topic in topics:  
    summary = chain.invoke({"input": topic})  
    print(f"Topic: {topic} | Summary: {summary}")
```