THE FUTURE OF AI & DEVELOPMENT

Module 8 — Harwell Prompt Engineering

LEARNING OBJECTIVES

By the end of this module you will be able to:

-
-
-
-

LEARNING OBJECTIVES

By the end of this module you will be able to:

- Summarise recent developments (e.g. last ~6 months): reasoning models, agentic workflows
-
-
-

LEARNING OBJECTIVES

By the end of this module you will be able to:

- Summarise recent developments (e.g. last ~6 months): reasoning models, agentic workflows
- Anticipate what’s coming next for Java developers: tooling, practices, and architectural patterns
-
-

LEARNING OBJECTIVES

By the end of this module you will be able to:

- Summarise recent developments (e.g. last ~6 months): reasoning models, agentic workflows
- Anticipate what’s coming next for Java developers: tooling, practices, and architectural patterns
- Place the course in context: what to try now vs. what to watch
-

LEARNING OBJECTIVES

By the end of this module you will be able to:

- Summarise recent developments (e.g. last ~6 months): reasoning models, agentic workflows
- Anticipate what’s coming next for Java developers: tooling, practices, and architectural patterns
- Place the course in context: what to try now vs. what to watch
- Leave with a simple “next steps” plan (e.g. one habit or tool to adopt)

BRIDGE FROM MODULE 7

What we learned last time:

-

The context:

-
-
-

Today: Recent developments, near-term trends, and concrete next steps.

BRIDGE FROM MODULE 7

What we learned last time:

- **How** AI APIs work (stateful, non-deterministic, streaming, cost)

The context:

-
-
-

Today: Recent developments, near-term trends, and concrete next steps.

BRIDGE FROM MODULE 7

What we learned last time:

- **How** AI APIs work (stateful, non-deterministic, streaming, cost)

The context:

- AI is moving fast
-
-

Today: Recent developments, near-term trends, and concrete next steps.

BRIDGE FROM MODULE 7

What we learned last time:

- **How** AI APIs work (stateful, non-deterministic, streaming, cost)

The context:

- AI is moving fast
- Let’s understand what’s changed recently
-

Today: Recent developments, near-term trends, and concrete next steps.

BRIDGE FROM MODULE 7

What we learned last time:

- **How** AI APIs work (stateful, non-deterministic, streaming, cost)

The context:

- AI is moving fast
- Let’s understand what’s changed recently
- What’s coming next

Today: Recent developments, near-term trends, and concrete next steps.

RECENT DEVELOPMENTS: LAST 6 MONTHS

Reasoning models:

RECENT DEVELOPMENTS: LAST 6 MONTHS

Reasoning models:

- Models that show step-by-step reasoning (e.g. o1-style)
-
-
-

RECENT DEVELOPMENTS: LAST 6 MONTHS

Reasoning models:

- Models that show step-by-step reasoning (e.g. o1-style)
- ✅ Better accuracy
-
-

RECENT DEVELOPMENTS: LAST 6 MONTHS

Reasoning models:

- Models that show step-by-step reasoning (e.g. o1-style)
- ✅ Better accuracy
- ✅ Can verify reasoning process
-

RECENT DEVELOPMENTS: LAST 6 MONTHS

Reasoning models:

- Models that show step-by-step reasoning (e.g. o1-style)
- ✅ Better accuracy
- ✅ Can verify reasoning process
- **Impact**: More reliable code generation, better explanations

Agentic workflows:

-
-
-

Agentic workflows:

- Models that plan → execute → reflect
-
-

Agentic workflows:

- Models that plan → execute → reflect
- ✅ Can handle multi-step tasks autonomously
-

Agentic workflows:

- Models that plan → execute → reflect
- ✅ Can handle multi-step tasks autonomously
- **Impact**: AI can plan refactoring, execute steps, verify results

What this means:

-
-
-
-
-

What this means:

- ✅ More reliable AI assistance
-
-
-
-

What this means:

- ✅ More reliable AI assistance
- ✅ AI can handle complex, multi-step tasks
-
-
-

What this means:

- ✅ More reliable AI assistance
- ✅ AI can handle complex, multi-step tasks
- ✅ Better explanations and reasoning
-
-

What this means:

- ✅ More reliable AI assistance
- ✅ AI can handle complex, multi-step tasks
- ✅ Better explanations and reasoning
- ⚠️ Still need human oversight
-

What this means:

- ✅ More reliable AI assistance
- ✅ AI can handle complex, multi-step tasks
- ✅ Better explanations and reasoning
- ⚠️ Still need human oversight
- ⚠️ Still need to verify output

NEAR-TERM: WHAT’S COMING NEXT

Trend 1: Better IDE integration

•

•

•

•

NEAR-TERM: WHAT’S COMING NEXT

Trend 1: Better IDE integration

- More AI features built into IDEs
-
-
-

NEAR-TERM: WHAT’S COMING NEXT

Trend 1: Better IDE integration

- More AI features built into IDEs
- Better context awareness
-
-

NEAR-TERM: WHAT'S COMING NEXT

Trend 1: Better IDE integration

- More AI features built into IDEs
- Better context awareness
- Smoother workflows
-

NEAR-TERM: WHAT’S COMING NEXT

Trend 1: Better IDE integration

- More AI features built into IDEs
- Better context awareness
- Smoother workflows
- **Impact**: Less switching between tools

Trend 2: More enterprise controls

-
-
-
-

Trend 2: More enterprise controls

- Better data privacy controls
-
-
-

Trend 2: More enterprise controls

- Better data privacy controls
- More governance features
-
-

Trend 2: More enterprise controls

- Better data privacy controls
- More governance features
- Compliance tools
-

Trend 2: More enterprise controls

- Better data privacy controls
- More governance features
- Compliance tools
- **Impact**: Safer AI adoption in enterprises

Trend 3: RAG/MCP becoming mainstream

-
-
-
-

Trend 3: RAG/MCP becoming mainstream

- RAG-enhanced tools more common
-
-
-

Trend 3: RAG/MCP becoming mainstream

- RAG-enhanced tools more common
- MCP adoption growing
-
-

Trend 3: RAG/MCP becoming mainstream

- RAG-enhanced tools more common
- MCP adoption growing
- Better integration with knowledge bases
-

Trend 3: RAG/MCP becoming mainstream

- RAG-enhanced tools more common
- MCP adoption growing
- Better integration with knowledge bases
- **Impact**: AI knows your context better

Trend 4: Impact on Java/Spring ecosystems

-
-
-
-

Trend 4: Impact on Java/Spring ecosystems

- Better Spring Boot code generation
-
-
-

Trend 4: Impact on Java/Spring ecosystems

- Better Spring Boot code generation
- Improved understanding of Java patterns
-
-

Trend 4: Impact on Java/Spring ecosystems

- Better Spring Boot code generation
- Improved understanding of Java patterns
- Better refactoring assistance
-

Trend 4: Impact on Java/Spring ecosystems

- Better Spring Boot code generation
- Improved understanding of Java patterns
- Better refactoring assistance
- **Impact**: More effective AI assistance for Java developers

BROADER LANDSCAPE: THEMES IN THE FUTURE OF AI

Security, sustainability, edge deployment, and more capable and efficient models will shape how we build and run software.

SECURITY OF AI AGENTS

SECURITY OF AI AGENTS

- AI systems are becoming more autonomous and pervasive
-
-

SECURITY OF AI AGENTS

- AI systems are becoming more autonomous and pervasive
- Need for robust security: malicious use, vulnerabilities, data protection
-

SECURITY OF AI AGENTS

- AI systems are becoming more autonomous and pervasive
- Need for robust security: malicious use, vulnerabilities, data protection
- **For developers:** Secure design, access controls, and oversight matter more as agents do more

AGI (ARTIFICIAL GENERAL INTELLIGENCE)

AGI (ARTIFICIAL GENERAL INTELLIGENCE)

- Long-term goal: AI with human-level cognitive abilities across any intellectual task
-
-

AGI (ARTIFICIAL GENERAL INTELLIGENCE)

- Long-term goal: AI with human-level cognitive abilities across any intellectual task
- Understand, learn, and apply intelligence broadly — not just narrow domains
-

AGI (ARTIFICIAL GENERAL INTELLIGENCE)

- Long-term goal: AI with human-level cognitive abilities across any intellectual task
- Understand, learn, and apply intelligence broadly — not just narrow domains
- **For developers:** A defining milestone to watch; today we work with narrow, capable models

SUSTAINABILITY

SUSTAINABILITY

- Environmental and ethical impact of AI: energy, compute, and resource use
-
-

SUSTAINABILITY

- Environmental and ethical impact of AI: energy, compute, and resource use
- Pressure for energy-efficient models and responsible deployment
-

SUSTAINABILITY

- Environmental and ethical impact of AI: energy, compute, and resource use
- Pressure for energy-efficient models and responsible deployment
- **For developers:** Efficiency and sustainability will influence tooling and architecture choices

AI AT THE EDGE

AI AT THE EDGE

- Processing AI workloads on devices and at the data source — not only in the cloud
-
-

AI AT THE EDGE

- Processing AI workloads on devices and at the data source — not only in the cloud
- Real-time applications, privacy, and lower latency
-

AI AT THE EDGE

- Processing AI workloads on devices and at the data source — not only in the cloud
- Real-time applications, privacy, and lower latency
- **For developers:** More options for where and how AI runs (devices, IoT, on-prem)

NEURO-SYMBOLIC AI

-
-
-

NEURO-SYMBOLIC AI

- Combines neural networks (pattern recognition) with symbolic AI (reasoning, knowledge)
-
-

NEURO-SYMBOLIC AI

- Combines neural networks (pattern recognition) with symbolic AI (reasoning, knowledge)
- Aim: more robust, explainable, human-like reasoning
-

NEURO-SYMBOLIC AI

- Combines neural networks (pattern recognition) with symbolic AI (reasoning, knowledge)
- Aim: more robust, explainable, human-like reasoning
- **For developers:** Could improve interpretability and reliability of AI-assisted tools

COMPACT, DOMAIN-SPECIFIC MODELS

COMPACT, DOMAIN-SPECIFIC MODELS

- Smaller, efficient models tailored for specific applications or narrow domains
-
-

COMPACT, DOMAIN-SPECIFIC MODELS

- Smaller, efficient models tailored for specific applications or narrow domains
- Contrast with large, general-purpose models; better for resource-constrained environments
-

COMPACT, DOMAIN-SPECIFIC MODELS

- Smaller, efficient models tailored for specific applications or narrow domains
- Contrast with large, general-purpose models; better for resource-constrained environments
- **For developers:** More deployable, cost-effective options for specialized tasks

WHAT TO TRY NOW VS. WHAT TO WATCH

Try now (practical focus):

WHAT TO TRY NOW VS. WHAT TO WATCH

Try now (practical focus):

- ✅ **Prompt engineering**: Apply what you learned (3Cs, iterative refinement)
-
-
-
-

WHAT TO TRY NOW VS. WHAT TO WATCH

Try now (practical focus):

- ✅ **Prompt engineering**: Apply what you learned (3Cs, iterative refinement)
- ✅ **Code generation**: Use for boilerplate, entities, services
-
-
-

WHAT TO TRY NOW VS. WHAT TO WATCH

Try now (practical focus):

- ✅ **Prompt engineering**: Apply what you learned (3Cs, iterative refinement)
- ✅ **Code generation**: Use for boilerplate, entities, services
- ✅ **Code explanation**: Understand legacy code before changing
-
-

WHAT TO TRY NOW VS. WHAT TO WATCH

Try now (practical focus):

- ✅ **Prompt engineering**: Apply what you learned (3Cs, iterative refinement)
- ✅ **Code generation**: Use for boilerplate, entities, services
- ✅ **Code explanation**: Understand legacy code before changing
- ✅ **Safe refactoring**: With tests, review carefully
-

WHAT TO TRY NOW VS. WHAT TO WATCH

Try now (practical focus):

- ✅ **Prompt engineering**: Apply what you learned (3Cs, iterative refinement)
- ✅ **Code generation**: Use for boilerplate, entities, services
- ✅ **Code explanation**: Understand legacy code before changing
- ✅ **Safe refactoring**: With tests, review carefully
- ✅ **Tool optimization**: Sidecar or integrated workflows

Watch (stay informed, don’t chase):

-
-
-
-

Watch (stay informed, don’t chase):

- ⚠️ **Reasoning models**: Watch adoption, try when stable
-
-
-

Watch (stay informed, don’t chase):

- ⚠️ **Reasoning models**: Watch adoption, try when stable
- ⚠️ **Agentic workflows**: Monitor, understand implications
-
-

Watch (stay informed, don’t chase):

- ⚠️ **Reasoning models**: Watch adoption, try when stable
- ⚠️ **Agentic workflows**: Monitor, understand implications
- ⚠️ **New tools**: Evaluate, don’t adopt immediately
-

Watch (stay informed, don’t chase):

- ⚠️ **Reasoning models**: Watch adoption, try when stable
- ⚠️ **Agentic workflows**: Monitor, understand implications
- ⚠️ **New tools**: Evaluate, don’t adopt immediately
- ⚠️ **Latest models**: Stay informed, but don’t chase every release

Avoid:

●

●

●

●

Avoid:

- ❌ Chasing every new release
-
-
-

Avoid:

- ❌ Chasing every new release
- ❌ Adopting bleeding-edge without evaluation
-
-

Avoid:

- ❌ Chasing every new release
- ❌ Adopting bleeding-edge without evaluation
- ❌ Ignoring fundamentals (prompting, review, testing)
-

Avoid:

- ❌ Chasing every new release
- ❌ Adopting bleeding-edge without evaluation
- ❌ Ignoring fundamentals (prompting, review, testing)
- ❌ Over-relying on AI without understanding

WHAT TO BE CAUTIOUS ABOUT: DANGEROUS CONFLUENCES

AI systems become especially risky when several capabilities combine. Worth regulating or constraining:

-
-

WHAT TO BE CAUTIOUS ABOUT: DANGEROUS CONFLUENCES

AI systems become especially risky when several capabilities combine. Worth regulating or constraining:

- **Self-improvement and self-direction** — Ability to modify their own code, goals, or behavior without human approval
-

WHAT TO BE CAUTIOUS ABOUT: DANGEROUS CONFLUENCES

AI systems become especially risky when several capabilities combine. Worth regulating or constraining:

- **Self-improvement and self-direction** — Ability to modify their own code, goals, or behavior without human approval
- **Control of resources** — Ability to acquire and control compute, money, or infrastructure

- **Replication** — Ability to create copies of themselves or deploy new instances
-

- **Replication** — Ability to create copies of themselves or deploy new instances
- **Persistence and evasion of shutdown** — Ability to maintain presence or resist being turned off

-

When these combine, the case for oversight and regulation strengthens.

- **Strategic, long-horizon autonomy** — Pursuing goals over long action sequences without human-in-the-loop

When these combine, the case for oversight and regulation strengthens.

COURSE RECAP: JOURNEY OVERVIEW

Module 1: Where/when to use AI safely (foundations)

Module 2: How to prompt effectively (core skill)

Module 3: What to prompt for in Java (practical application)

Module 4: Tooling strategies (workflow optimization)

Module 5: RAG (connecting to knowledge)

Module 6: MCP (connecting to systems)

Module 7: AI APIs (technical details)

Module 8: Future (what’s next)

Key themes:

-
-
-
-
-
-
-

Key themes:

- Safety first
-
-
-
-
-
-

Key themes:

- Safety first
- Prompting is a skill
-
-
-
-
-

Key themes:

- Safety first
- Prompting is a skill
- Practical Java development
-
-
-
-

Key themes:

- Safety first
- Prompting is a skill
- Practical Java development
- Optimize your workflow
-
-
-

Key themes:

- Safety first
- Prompting is a skill
- Practical Java development
- Optimize your workflow
- Connect AI to your context
-
-

Key themes:

- Safety first
- Prompting is a skill
- Practical Java development
- Optimize your workflow
- Connect AI to your context
- Understand the technology
-

Key themes:

- Safety first
- Prompting is a skill
- Practical Java development
- Optimize your workflow
- Connect AI to your context
- Understand the technology
- Stay informed, focus on fundamentals

NEXT STEPS: WHAT MUST FEEL DIFFERENT ON MONDAY

The challenge:

-
-
-

NEXT STEPS: WHAT MUST FEEL DIFFERENT ON MONDAY

The challenge:

- “You’ve learned a lot. What will you actually do differently?”
-
-

NEXT STEPS: WHAT MUST FEEL DIFFERENT ON MONDAY

The challenge:

- “You’ve learned a lot. What will you actually do differently?”
- ❌ Easy to forget without a plan
-

NEXT STEPS: WHAT MUST FEEL DIFFERENT ON MONDAY

The challenge:

- “You’ve learned a lot. What will you actually do differently?”
- ❌ Easy to forget without a plan
- ❌ Hard to change habits

The solution:

One habit to adopt:

-

-

-

The solution:

One habit to adopt:

- Example: “Use few-shot prompting for all code generation”
-
-

The solution:

One habit to adopt:

- Example: “Use few-shot prompting for all code generation”
- Example: “Always include context header in prompts”
-

The solution:

One habit to adopt:

- Example: “Use few-shot prompting for all code generation”
- Example: “Always include context header in prompts”
- Example: “Review AI output with evaluation checklist”

One tool to try:

-

-

One tool to try:

- Example: “Optimize sidecar workflow with context headers”
-

One tool to try:

- Example: “Optimize sidecar workflow with context headers”
- Example: “Try integrated IDE features”

One prompt template:

-
-

One prompt template:

- Example: Context header template
-

One prompt template:

- Example: Context header template
- Example: Code generation prompt template

Commitment:

-
-
-

Commitment:

- Write it down
-
-

Commitment:

- Write it down
- Share with partner or team
-

Commitment:

- Write it down
- Share with partner or team
- Set a reminder to review in one week

CLOSING: KEY TAKEAWAYS

Fundamentals matter:

-
-
-

CLOSING: KEY TAKEAWAYS

Fundamentals matter:

- Prompting principles don’t change
-
-

CLOSING: KEY TAKEAWAYS

Fundamentals matter:

- Prompting principles don’t change
- Review practices remain important
-

CLOSING: KEY TAKEAWAYS

Fundamentals matter:

- Prompting principles don’t change
- Review practices remain important
- Testing is still essential

Stay informed, don’t chase:

-
-
-

Stay informed, don’t chase:

- Understand trends
-
-

Stay informed, don’t chase:

- Understand trends
- Adopt when ready
-

Stay informed, don’t chase:

- Understand trends
- Adopt when ready
- Focus on practical skills

Practical focus:

-
-

Practical focus:

- What to try now: Prompt engineering, code generation, safe refactoring
-

Practical focus:

- What to try now: Prompt engineering, code generation, safe refactoring
- What to watch: Reasoning models, agentic workflows, new tools

Next steps:

-
-
-

Next steps:

- One habit
-
-

Next steps:

- One habit
- One tool
-

Next steps:

- One habit
- One tool
- One template

FINAL MESSAGE

AI is a tool. You’re the developer. Use it wisely.

-
-
-

FINAL MESSAGE

AI is a tool. You’re the developer. Use it wisely.

- Focus on fundamentals: prompting, review, testing
-
-

FINAL MESSAGE

AI is a tool. You’re the developer. Use it wisely.

- Focus on fundamentals: prompting, review, testing
- Stay informed, but don’t chase every release
-

FINAL MESSAGE

AI is a tool. You’re the developer. Use it wisely.

- Focus on fundamentals: prompting, review, testing
- Stay informed, but don’t chase every release
- What must feel different on Monday? You have a plan.

QUESTIONS?

Thank you for participating!

Module 8 — The Future of AI & Development