**Amazon Q Developer Training - Day 1 Trainer Handbook**

**Day 1 Theme:** Getting Started with Amazon Q Developer

### ✅ Session Objectives:

* Understand Amazon Q Developer’s capabilities and role in GenAI-based software development.
* Install and configure Amazon Q Developer in VS Code and/or Cloud9.
* Explore Q’s conversational interface, code generation, and basic prompts.
* Run your first program using Q assistance.

### ⏰ Session Flow:

#### 1. Welcome & Introduction (15 mins)

* Welcome participants and give an overview of the 4-day agenda.
* Explain how GenAI is transforming software development.
* Highlight Amazon Q vs ChatGPT/Copilot.

**Trainer Script:** “Amazon Q Developer is not just a code assistant. It’s your productivity partner integrated tightly with AWS and your IDE. Today, we’ll set it up and explore its capabilities.”

#### 2. What is Amazon Q Developer? (20 mins)

* Explain Amazon Q’s 3 offerings: Developer, Business, and Apps.
* Focus on Q Developer:
  + Generates and explains code
  + Finds and fixes bugs
  + Writes tests and documentation
  + Answers dev-related AWS questions

**Trainer Notes:** - Compare Q Developer with GitHub Copilot: AWS-native, deeper cloud integration - Show real use cases: microservices, Lambda, APIs

**Exercise:** Ask Q: “Explain this function” on a JavaScript or Java method

#### 3. Installing Amazon Q Developer (25 mins)

* Option 1: Install Q in **VS Code** (with AWS Toolkit)
* Option 2: Use **Cloud9 IDE** (Q is pre-installed)
* Sign in with AWS Builder ID

**Trainer Script:** “We’ll all install Q Developer now. If you’re using Cloud9, you’re ready to go. Otherwise, follow the VS Code extension method.”

**Lab:** - Guide learners to open a repo or new file - Ask Q: “Generate a Hello World program in Java” - Ask Q: “Create a REST API using Spring Boot”

#### 4. Exploring the Interface (20 mins)

* Conversational chat panel
* Code annotation sidebar
* Integration with AWS resources

**Demo:** - Type: “Create a Lambda function that stores data in S3” - Observe step-by-step generation

**Q Prompts:** - “Explain this class” - “Add logging to this method” - “Write unit test for this function”

#### 5. Hands-On Lab 1: Q in Action (30 mins)

* Choose language: Java or Node.js
* Create a simple app using Q prompts
* Ask Q to explain/refactor code

**Java Example Prompts:** - “Create a Spring Boot REST API with 2 endpoints” - “Add exception handling” - “Write JUnit tests”

**Node.js Example Prompts:** - “Create an Express server with /login route” - “Add MongoDB integration”

#### 6. Recap + Q&A (10 mins)

* Recap key capabilities explored today
* Highlight how Q saves time and effort
* Share Day 2 topics: advanced generation, navigating legacy code

**Take-Home:** - Install Q on personal system - Try 3 custom prompts with Q - Observe code quality and explanations

### 🎒 Trainer Tips:

* Keep examples short and clear
* Allow learners to experiment with prompts
* Reinforce security and correctness review

### ✍️ Suggested Pre-Work for Day 2:

* Explore Amazon Q documentation
* Read sample repo with legacy code (to be explained using Q)
* Note areas where Q could assist

**Amazon Q Developer Training - Day 2 Trainer Handbook**

**Day 2 Theme:** Working with Legacy Code and Advanced Prompts

### ✅ Session Objectives:

* Learn how to use Q to navigate, understand, and refactor legacy code.
* Utilize advanced prompt engineering to enhance code quality.
* Explore Q’s understanding of large codebases.
* Use Q to write integration tests and documentation.

### ⏰ Session Flow:

#### 1. Warm-Up & Recap (10 mins)

* Quick recap of Day 1 learnings
* Share observations and prompt results from take-home task

#### 2. Exploring Legacy Code with Q (30 mins)

* Open a legacy Java or Node.js repo
* Ask Q to explain architecture, key functions, and dependencies
* Ask Q to locate a bug or optimization opportunity

**Q Prompts:** - “What does this controller do?” - “Where is the entry point of this app?” - “Find places using deprecated methods”

#### 3. Refactoring Legacy Code (20 mins)

* Use Q to simplify complex methods
* Convert monolith code into service functions

**Lab:** - Use Q to identify code smells - Prompt Q: “Refactor this class to improve readability”

#### 4. Writing Integration Tests with Q (20 mins)

* Generate test cases with Q for real-world classes/functions
* Discuss test coverage and validation

**Demo:** - Ask Q: “Write JUnit tests for this service” - Ask Q: “Write test cases using Jest for Express route”

#### 5. Documenting with Q (15 mins)

* Use Q to generate README, method comments, and diagrams

**Prompt Examples:** - “Generate README for this project” - “Add Javadoc to all public methods”

#### 6. Lab 2: Full Q Audit (30 mins)

* Task: Audit and refactor a provided legacy app (Java/Node.js)
* Add missing docs and test cases

#### 7. Recap + Homework (10 mins)

* Summary: Legacy navigation, refactoring, test/document generation

**Homework:** - Use Q on your own projects - Prepare to explore AWS with Q on Day 3

**Amazon Q Developer Training - Day 3 Trainer Handbook**

**Day 3 Theme:** Amazon Q with AWS Services

### ✅ Session Objectives:

* Learn how Amazon Q integrates with AWS services
* Generate AWS infrastructure code (IaC)
* Use Q to create Lambda, S3, DynamoDB apps
* Review Q’s cloud awareness and AWS-specific guidance

### ⏰ Session Flow:

#### 1. Recap & Introduction (10 mins)

* Review Day 2 topics
* Explain today’s AWS-focused hands-on work

#### 2. AWS Services via Q (25 mins)

* Prompt Q to create and configure:
  + Lambda
  + API Gateway
  + S3
  + DynamoDB

**Prompts:** - “Create Lambda to process JSON” - “Upload file to S3 from form” - “Store user data in DynamoDB”

#### 3. Q for Infrastructure as Code (30 mins)

* Generate CDK or CloudFormation using Q
* Customize and deploy

**Demo:** - Prompt: “Generate CDK app for Lambda + S3” - Explain deployment process

#### 4. Lab 3: Build & Deploy with Q (40 mins)

* Task: Use Q to build and deploy a sample AWS app
* Include REST API, Lambda, and DB

#### 5. Security & Cost Awareness in Q (15 mins)

* Q’s recommendations around IAM roles
* Cost-effective design prompts

#### 6. Recap + Assignment (10 mins)

* Summary: Q’s AWS integration, IaC generation

**Assignment:** - Use Q to deploy something small on AWS - Capture screenshots or code

**Amazon Q Developer Training - Day 4 Trainer Handbook**

**Day 4 Theme:** Real-World Scenarios, DevOps & Productivity Boosts

### ✅ Session Objectives:

* Use Q for DevOps tasks, CI/CD, debugging
* Speed up review and documentation
* Build a project using Q from scratch
* Final evaluation and certification

### ⏰ Session Flow:

#### 1. Recap & Motivation (10 mins)

* Quick review of AWS integration (Day 3)
* Set Day 4 goal: Project + DevOps with Q

#### 2. Q in DevOps (25 mins)

* Prompts for Docker, CI/CD pipelines, GitHub Actions

**Examples:** - “Create Dockerfile for Spring Boot app” - “Create GitHub Actions to test and deploy”

#### 3. Code Review & Debugging with Q (25 mins)

* Ask Q to analyze pull requests
* Debug with Q explanations

#### 4. Final Project Lab (60 mins)

* Build an end-to-end app using Q
* Use DevOps tools, documentation, AWS integration

#### 5. Presentations & Feedback (20 mins)

* Participants demo their projects
* Trainer gives feedback and suggestions

#### 6. Wrap-Up & Certification (10 mins)

* Review 4-day learning outcomes
* Share feedback link
* Distribute certificates

🎓 **Post-Training Resources:** - Amazon Q Docs & FAQs - GitHub sample projects - AWS Free Tier signup links - Community links for continued learning