



Apprenticeship Program Occupational Standards

Application Developer on IBM Z Competency Framework

O*NET CODE: 15-1132.00 (Software Developers, Application)
Updated on 3/1/2023

This document was created by IBM's Apprenticeship Program as an open source standard to help industry accelerate their journey to developing new collar apprenticeship and work-based learning programs.

Contents

Table of Contents

Work Process Schedule and Related Instruction Outline	3
Overview	3
Work Processes	4
Competencies and Performance Criteria	5
Evidence Types	7
Foundational Performance Criteria.....	8
Outline of Related Instruction	18
Professional Foundational Instruction	18
Technical Foundational Instruction.....	18

Work Process Schedule and Related Instruction Outline

Overview

The occupational standards include the competency framework that outlines the required technical and professional competencies for each occupation. Demonstrated proficiency in all competencies is a requirement for successful completion of a competency-based apprenticeship program.

The following section contains work process, training outline, and related instruction for the **IBM Application Developer on IBM Z Apprenticeship**. The criteria are segregated into three distinct parts.

- **Part I: Work Process** – This section delineates the general outline of basic, high-level requirements that each participant will need to satisfy including projects, coaching, job shadowing, and training.
- **Part II: Competencies and Performance Criteria** – In support of this competency-based apprenticeship model, this section identifies what technical knowledge and professional behaviors will be evident as a product of achieving proficiency in these areas.
 - Competency Outcomes – These are the skills that apprentices are required to master during their apprenticeship. Competencies are defined as knowledge (K), behavior (B), or skills (S).
 - Assessment Criteria - Outlines the specific knowledge or combination of skills that each apprentice is required to learn to demonstrate proficiency.
 - Evidence types are the mechanisms used to evaluate apprentice's overall proficiency in a stated competency.
- **Part III: Outline of Related Instruction** – This section outlines specific formal training that each participant will be required to complete or demonstrate mastery.

Work Processes

On the Job Training:	Validated by Manager / Mentor	Date
Principles and Practices		
1. Apply agile principles and practices		
2. Leverage IBM Design Thinking practices		
3. Apply business and professional acumen skills		
4. Use and contribute to shared and open GitHub repositories		
5. Gather and analyze data to draw insights		
Software Engineering Fundamentals		
6. Conduct software design and modeling		
7. Develop and write software code		
8. Perform software testing and problem solving		
9. Perform system scaling and security		
DevOps Fundamentals		
10. Perform continuous integration of code		
11. Perform continuous delivery of code		
12. Execute automation as part of the development lifecycle		
13. Implement metrics and measurement		
Platforms, Services, and Solutions		
14. Use various cloud tools, services, and platforms		
15. Use logging and monitoring tools		
16. Migrate data from on-premise solution to cloud solution		

Competencies and Performance Criteria

Foundational Competencies

#	Knowledge / Skill / Behavior	Description
Principles and Practices		
1.0	K / S / B	Understand, articulate, and demonstrate agile principles and practices
2.0	K / S / B	Understand, articulate, and demonstrate Design Thinking
3.0	B	Demonstrate key teamwork and collaborative behaviors
4.0	B	Demonstrate strong communication skills
5.0	K / B	Understand and model good feedback behaviors
6.0	K / B	Understand and demonstrate social coding behaviors
7.0	K / S	Demonstrate ability to analyze data sets, identify insights, and leverage to drive decision making
Software Engineering Fundamentals		
8.0	K	Demonstrate knowledge of key computer programming fundamentals
9.0	K / S	Understand and demonstrate key software design fundamentals
10.0	K / S	Understand and demonstrate test-driven development
11.0	K	Understand and navigate the complexity associated with enterprise-level development
12.0	K / S / B	Understand, articulate, and demonstrate clean coding behaviors
13.0	K / B	Understand and manage technical debt
14.0	K / S	Understand and demonstrate knowledge of web programming skills
15.0	K / S	Understand how to use version control for all elements of the software delivery lifecycle
16.0	K / S	Understand and demonstrate how to construct and test quality code, at scale
DevOps Fundamentals		
17.0	K / S	Understand and demonstrate continuous integration
18.0	K / S	Understand and demonstrate continuous delivery
19.0	K / S	Understand and demonstrate feature decoupling
20.0	K / S	Understand and demonstrate DevOps automation
21.0	K / S	Understand and demonstrate DevOps metrics and measurements
Platforms, Services, and Solutions		
22.0	K / S	Understand and demonstrate knowledge of cloud computing fundamentals, including the various tools, services, and principles
23.0	K / S	Understand and demonstrate the design patterns and practices for building cloud native services
24.0	K / S	Understand the relationship between scaling techniques, how to exploit them, and where

25.0	K	Understand the various platforms, their differences, relative strengths / weaknesses, and integrations
26.0	K	Understand the topologies of enterprise solutions, and how clients use our portfolio of products and services together
27.0	K / S	Understand and demonstrate the use of logging and monitoring tools and infrastructure
28.0	K	Understand the need for mobility and migration of data from on-premise to cloud solutions, and the implications
29.0	K / S	Understand and demonstrate knowledge of Enterprise computing platform
30.0	K / S	Demonstrate basic understanding of the hierarchical and relational DBMS
31.0	K / S	Understand the concepts, functions and facilities of CICS

Evidence Types

Evidence Code	Description
O	Observation
Q&A	Questions and answers
P	Learner products
RA	Reflective accounts / personal statements
S	Simulation
PD	Professional discussion
A	Assignments, projects, case studies
MT	Mentor testimony
EW	Expert witness evidence
RPL	Recognition of prior learning

Foundational Performance Criteria

Job Role	Application Developer on IBM Z Specialist
O*NET Code	15.1132 (Software Developers, Application)
Apprenticeship Level	Foundation
Guided Learning Hours	338
Experiential Hours	1662

Competency Outcome	Assessment Criteria	Evidence Types	Education Material
1.0 Understand, articulate, and demonstrate agile principles and practices	1.1 Demonstrate knowledge of agile principles and how IBM implements them in a development environment 1.2 Lead and participate in agile planning activities, including story points, planning poker, Kanban boards, prioritizing work, and writing stories 1.3 Model “small batch” practices by successfully breaking down work tasks into smaller components 1.4 Lead and participate in retrospectives to drive continuous improvement	<input checked="" type="checkbox"/> O <input checked="" type="checkbox"/> Q&A <input type="checkbox"/> P <input type="checkbox"/> RA <input type="checkbox"/> S <input type="checkbox"/> PD <input checked="" type="checkbox"/> A <input checked="" type="checkbox"/> MT <input type="checkbox"/> EW <input checked="" type="checkbox"/> RPL	Agile Explorer Agile Fundamentals Agile in the Real World
2.0 Understand, articulate, demonstrate IBM Design Thinking	2.1 Demonstrate knowledge of IBM Design Thinking and how IBM implements it in a development environment 2.2 Create empathy maps for identified personas 2.3 Execute discovery phase to identify customer requirements 2.4 Develop hills that communicate project intent with clarity and flexibility 2.5 Conduct successful playbacks with stakeholders to exchange feedback and measure progress 2.6 Identify and leverage sponsor users to provide real-world perspective 2.7 Demonstrate behaviors and work plans aligned with fail fast / MVP principles	<input checked="" type="checkbox"/> O <input checked="" type="checkbox"/> Q&A <input checked="" type="checkbox"/> P <input checked="" type="checkbox"/> RA <input checked="" type="checkbox"/> S <input type="checkbox"/> PD <input checked="" type="checkbox"/> A <input checked="" type="checkbox"/> MT <input type="checkbox"/> EW <input checked="" type="checkbox"/> RPL	Enterprise Design Thinking Practitioner - Training
3.0 Demonstrate teamwork and collaborative behaviors	3.1 Demonstrate ability to work co-operative with others 3.2 Demonstrate ability to work as part of a team 3.3 Demonstrate successful use of playbacks and reviews 3.4 Demonstrate goal-setting, being solution-focused, managing and strengthening relationships, and working with diverse perspectives.	<input checked="" type="checkbox"/> O <input type="checkbox"/> Q&A <input type="checkbox"/> P <input checked="" type="checkbox"/> RA <input checked="" type="checkbox"/> S <input checked="" type="checkbox"/> PD <input checked="" type="checkbox"/> A <input checked="" type="checkbox"/> MT <input checked="" type="checkbox"/> EW <input checked="" type="checkbox"/> RPL	Agile Explorer

<p>4.0 Demonstrate strong communication skills</p>	<p>4.1 Demonstrate strong communication skills through the selection of the appropriate communication method for each message</p> <p>4.2 Demonstrate strong presentation skills through quality materials and clear presentation</p> <p>4.3 Demonstrate appropriate use of email for general communications</p> <p>4.4 Demonstrate appropriate use of Slack for general communications</p> <p>4.5 Demonstrate appropriate use of Slack as a collaboration tool</p>	<p><input checked="" type="checkbox"/> O</p> <p><input checked="" type="checkbox"/> Q&A</p> <p><input type="checkbox"/> P</p> <p><input checked="" type="checkbox"/> RA</p> <p><input type="checkbox"/> S</p> <p><input checked="" type="checkbox"/> PD</p> <p><input checked="" type="checkbox"/> A</p> <p><input checked="" type="checkbox"/> MT</p> <p><input checked="" type="checkbox"/> EW</p> <p><input checked="" type="checkbox"/> RPL</p>	<p>Professional Skills</p> <p>Leadership and Communication Skills for Software Engineers</p>
<p>5.0 Understand and model good feedback behaviors</p>	<p>5.1 Understand the importance of feedback in all we do</p> <p>5.2 Successfully leverage Net Promoter Score methodology</p> <p>5.3 Deliver quality feedback to team members</p> <p>5.4 Receive feedback gracefully and act on it</p> <p>5.5 Model goal setting behaviors through performance management system</p> <p>5.6 Model a culture of feedback with all team members</p>	<p><input checked="" type="checkbox"/> O</p> <p><input checked="" type="checkbox"/> Q&A</p> <p><input type="checkbox"/> P</p> <p><input type="checkbox"/> RA</p> <p><input checked="" type="checkbox"/> S</p> <p><input checked="" type="checkbox"/> PD</p> <p><input checked="" type="checkbox"/> A</p> <p><input checked="" type="checkbox"/> MT</p> <p><input checked="" type="checkbox"/> EW</p> <p><input checked="" type="checkbox"/> RPL</p>	<p>Professional Skills</p>
<p>6.0 Understand and demonstrate social coding behaviors</p>	<p>6.1 Demonstrate knowledge of social coding behaviors</p> <p>6.2 Execute successful "Fork and Pull" model in GitHub -</p> <p>6.3 Demonstrate effective searching via GitHub / GHE to identify code</p> <p>6.4 Leverage established channels, i.e. slack, to collaborate and identify code</p> <p>6.5 Explain the use of Open Source and the implications</p> <p>6.6 Demonstrate successful usage of Open Sources scanning tools</p> <p>6.7 Articulate the appropriate protection and usage of Intellectual Property</p>	<p><input checked="" type="checkbox"/> O</p> <p><input checked="" type="checkbox"/> Q&A</p> <p><input type="checkbox"/> P</p> <p><input type="checkbox"/> RA</p> <p><input checked="" type="checkbox"/> S</p> <p><input checked="" type="checkbox"/> PD</p> <p><input checked="" type="checkbox"/> A</p> <p><input checked="" type="checkbox"/> MT</p> <p><input checked="" type="checkbox"/> EW</p> <p><input checked="" type="checkbox"/> RPL</p>	<p>Clean Coding Principles in C#</p> <p>GitHub: Getting Started</p> <p>Inspecting Open Source Software for Security and License Compliance</p>
<p>7.0 Demonstrate ability to analyze data sets, identify insights, and leverage to drive decision making</p>	<p>7.1 Understand the value of real-time data gathering to reduce risk and increase success</p> <p>7.2 Articulate the different types of analytics: predictive, prescriptive, descriptive, cognitive, machine learning</p> <p>7.3 Demonstrate use of data analytics to make decisions</p> <p>7.4 Successfully analyze large data sets and leverage data to drive insights</p>	<p><input checked="" type="checkbox"/> O</p> <p><input checked="" type="checkbox"/> Q&A</p> <p><input type="checkbox"/> P</p> <p><input type="checkbox"/> RA</p> <p><input type="checkbox"/> S</p> <p><input checked="" type="checkbox"/> PD</p> <p><input checked="" type="checkbox"/> A</p> <p><input checked="" type="checkbox"/> MT</p> <p><input checked="" type="checkbox"/> EW</p> <p><input type="checkbox"/> RPL</p>	<p>Data Analytics hands-on</p> <p>Big Data 101</p> <p>Hadoop 101</p> <p>Spark Fundamentals</p>

<p>8.0 Demonstrate knowledge of key computer programming fundamentals</p>	<p>8.1 Demonstrate an understanding of the history of computing</p> <p>8.2 Demonstrate an understanding of the history and principles of software engineering</p> <p>8.3 Demonstrate an understanding of the history and principles of data management</p> <p>8.4 Demonstrate an understanding of the history and principles of networking fundamentals</p> <p>8.5 Demonstrate an understanding of the history and principles of infrastructure</p> <p>8.6 Demonstrate an understanding of the history and principles of integrated development environments</p>	<p><input checked="" type="checkbox"/> O</p> <p><input checked="" type="checkbox"/> Q&A</p> <p><input type="checkbox"/> P</p> <p><input checked="" type="checkbox"/> RA</p> <p><input type="checkbox"/> S</p> <p><input type="checkbox"/> PD</p> <p><input type="checkbox"/> A</p> <p><input checked="" type="checkbox"/> MT</p> <p><input type="checkbox"/> EW</p> <p><input checked="" type="checkbox"/> RPL</p>	<p>TCP/IP Networking for Developers</p> <p>AP College Computer Science Principles</p>
<p>9.0 Understand and demonstrate key software design fundamentals</p>	<p>9.1 Demonstrate knowledge and understanding of key software design fundamentals (for e.g. SDLC) - Sections 3.1 to 3.6 - Developer to Architect; further detail Software Engineering Essentials.</p> <p>9.2 Demonstrate successful problem-solving behaviors</p> <p>9.2.1 Demonstrate ability to decompose a problem</p> <p>9.2.2 Demonstrate ability to use abstract data types</p> <p>9.2.3 Demonstrate ability to apply algorithms as a solution</p> <p>9.2.4 Demonstrate ability to follow logical conclusions</p> <p>9.3 Demonstrate the use of an object-oriented language or modelling language to design a model that promotes clean code - Developer to Architect- Module 3</p> <p>9.3.1 Successfully decompose into discrete components in a coherent design that will enable the software to grow and be maintained efficiently - Developer to Architect</p> <p>9.4 Demonstrate the use of functional programming skills –</p> <p>9.5 Demonstrate ability to recognize and apply design patterns to address a problem – Developer to Architect</p> <p>9.6 Demonstrate proficiency in languages like COBOL, PL/1, Assembler, Java, Python, or JavaScript – demonstrated through a coding example</p>	<p><input checked="" type="checkbox"/> O</p> <p><input checked="" type="checkbox"/> Q&A</p> <p><input checked="" type="checkbox"/> P</p> <p><input type="checkbox"/> RA</p> <p><input checked="" type="checkbox"/> S</p> <p><input type="checkbox"/> PD</p> <p><input checked="" type="checkbox"/> A</p> <p><input checked="" type="checkbox"/> MT</p> <p><input type="checkbox"/> EW</p> <p><input type="checkbox"/> RPL</p>	<p>Java Fundamentals: The Java Language</p> <p>Python: The Big Picture</p> <p>Core Python 3.6: Getting Started</p> <p>COBOL Programming with VSCode</p> <p>Quick Start to JavaScript: Volume 1</p> <p>Quick Start to JavaScript: Volume 2</p> <p>Quick Start to JavaScript: Volume 3</p> <p>Java 8 Fundamentals: The Core Platform</p> <p>Java 11 Fundamentals: Collection</p>

			Building a JavaScript Development Environment Node.js 12: Getting Started Clean Coding Principles in C# Developer to Architect Software Engineering Essentials
10.0 Understand and demonstrate test-driven development	10.1 Articulate the differences and relative strengths / weakness of the various models of test-driven or behavior driven development 10.2 Demonstrate use of various models of test-driven or behavior driven development 10.3 Demonstrate understanding of testing life cycle and levels of testing (for e.g: Unit test, Function test (Green screen testing), System Test, Integration test, Regression test, etc)	<input checked="" type="checkbox"/> O <input checked="" type="checkbox"/> Q&A <input checked="" type="checkbox"/> P <input type="checkbox"/> RA <input checked="" type="checkbox"/> S <input type="checkbox"/> PD <input checked="" type="checkbox"/> A <input checked="" type="checkbox"/> MT <input type="checkbox"/> EW <input checked="" type="checkbox"/> RPL	Test Driven Development – The Big Picture
11.0 Understand and navigate the complexity associated with enterprise-level development	11.1 Articulate the complexities and challenges associated with enterprise-scale development - Course 2 Understanding Enterprise Architecture, Chapter 1, Module 3 "Making sense of plurality" 11.2 Understand how to apply security at enterprise - Course 2 Understanding Enterprise Architecture, Chapter 7, Module 4 "Security Architecture" 11.3 Understand how to scale solutions - Course 2 Understanding Enterprise Architecture, Chapter 4, Module 6 "Elastic scaling and Continuous Delivery" 11.4 Understand how to manage performance at enterprise - Course 2 Understanding Enterprise Architecture, Chapter 4, Module 6 "Elastic scaling and Continuous Delivery" 11.5 Understand how to address legacy integrations - Course 2 Understanding Enterprise Architecture, Chapter 4, Module 8 "Convergence"	<input checked="" type="checkbox"/> O <input checked="" type="checkbox"/> Q&A <input checked="" type="checkbox"/> P <input type="checkbox"/> RA <input checked="" type="checkbox"/> S <input type="checkbox"/> PD <input type="checkbox"/> A <input checked="" type="checkbox"/> MT <input type="checkbox"/> EW <input checked="" type="checkbox"/> RPL	Clean Architecture – Patterns, Practices & Principles Understanding Enterprise Architecture Overview of the TOGAF 9.1 Enterprise Architecture Framework
12.0 Understand, articulate, and demonstrate clean coding behaviors	12.1 Demonstrate use of refactoring to yield clean code 12.2 Demonstrate ability to write code with clarity 12.3 Demonstrate ability to write code that minimizes duplication 12.4 Demonstrate ability to write code with simplicity	<input checked="" type="checkbox"/> O <input checked="" type="checkbox"/> Q&A <input checked="" type="checkbox"/> P <input type="checkbox"/> RA <input checked="" type="checkbox"/> S	Clean Coding Principles in C#

	12.5 Explain the various clean coding principles represented by the acronym SOLID 12.6 Demonstrate use of SOLID principles in code	<input type="checkbox"/> PD <input type="checkbox"/> A <input checked="" type="checkbox"/> MT <input type="checkbox"/> EW <input checked="" type="checkbox"/> RPL	
13.0 Understand and manage technical debt	13.1 Describe the issues caused by technical debt - Course 2 Understanding and Eliminating Technical Debt, Module 1 13.2 Demonstrate ability to identify technical debt - Course 2 Understanding and Eliminating Technical Debt, Module 2 13.3 Demonstrate ability to eliminate technical debt - Course 2 Understanding and Eliminating Technical Debt, Module 5 & 6 13.4 Articulate and/or demonstrate ability to use code scanning and static analysis tools	<input checked="" type="checkbox"/> O <input checked="" type="checkbox"/> Q&A <input checked="" type="checkbox"/> P <input type="checkbox"/> RA <input type="checkbox"/> S <input type="checkbox"/> PD <input type="checkbox"/> A <input checked="" type="checkbox"/> MT <input type="checkbox"/> EW <input checked="" type="checkbox"/> RPL	Clean Coding Principles in C# Understanding and Eliminating Technical Debt
14.0 Understand and demonstrate knowledge of web programming skills	14.1 Understand and articulate the value of distributed, modern computing architecture and the role it plays in modern service development 14.2 Demonstrate proficient back-end or front-end development skills 14.3 Demonstrate basic knowledge of back-end and front-end development 14.4 Demonstrate understanding of how to scale applications using database 14.5 Demonstrate understanding of how to build and deploy scalable applications	<input checked="" type="checkbox"/> O <input checked="" type="checkbox"/> Q&A <input checked="" type="checkbox"/> P <input type="checkbox"/> RA <input type="checkbox"/> S <input type="checkbox"/> PD <input checked="" type="checkbox"/> A <input checked="" type="checkbox"/> MT <input type="checkbox"/> EW <input checked="" type="checkbox"/> RPL	Front-end Web Development Quick Start with HTML5, CSS, and JavaScript Introduction to Web Development jQuery Fundamentals Angular: Getting Started Designing RESTful Web APIs A Practical Start with React Building Applications with React 16 and Flux 3
15.0 Understand how to use version control for all elements of	15.1 Describe the principles of version control 15.2 Describe versioning, releases, issues, and merges in source code	<input checked="" type="checkbox"/> O <input checked="" type="checkbox"/> Q&A <input checked="" type="checkbox"/> P	OJT

the software delivery lifecycle	15.3 Demonstrate proper usage of versioning, releases, issues, and merges in source code 15.4 Understand the use of dependencies and libraries from within and outside the team 15.5 Demonstrate usage of dependencies and libraries from within and outside the team	<input type="checkbox"/> RA <input type="checkbox"/> S <input type="checkbox"/> PD <input checked="" type="checkbox"/> A <input checked="" type="checkbox"/> MT <input type="checkbox"/> EW <input checked="" type="checkbox"/> RPL	
16.0 Understand and demonstrate how to construct and test quality code, at scale	16.1 Understand how to construct quality code at scale 16.2 Understand the techniques used for ensuring quality 16.3 Understand the different types of testing, where to apply them, and the relative amounts 16.4 Understand the successful use of test coverage tools 16.5 Understand successful use of performance testing 16.6 Understand successful use of boundary and limit testing Understand successful use of chaotic testing	<input checked="" type="checkbox"/> O <input checked="" type="checkbox"/> Q&A <input checked="" type="checkbox"/> P <input type="checkbox"/> RA <input checked="" type="checkbox"/> S <input type="checkbox"/> PD <input checked="" type="checkbox"/> A <input checked="" type="checkbox"/> MT <input type="checkbox"/> EW <input checked="" type="checkbox"/> RPL	OJT
17.0 Understand and demonstrate continuous integration	17.1 Articulate the value of continuous integration in a development environment - Course 1 DevOps Foundations: Continuous Integration and Continuous Delivery Module 3 17.2 Demonstrate CI principles through frequent commits to trunk of codebase - Course 1, DevOps Foundations: Continuous Integration and Continuous Delivery Module 3 & 5 17.3 Demonstrate CI principles through use of automated tests to ensure codebase is in a good state - Course 1 DevOps Foundations: Continuous Integration and Continuous Delivery, Module 3 & 5 17.4 Demonstrate proficient use of continuous integration tools - Course 2 Getting Started with Jenkins and Course 3 Travis CI 17.5 Demonstrate understanding of the impact of commit defects into shared codebase - Course 1 DevOps Foundations: Continuous Integration and Continuous Delivery, Module 4	<input checked="" type="checkbox"/> O <input checked="" type="checkbox"/> Q&A <input checked="" type="checkbox"/> P <input type="checkbox"/> RA <input type="checkbox"/> S <input type="checkbox"/> PD <input type="checkbox"/> A <input checked="" type="checkbox"/> MT <input type="checkbox"/> EW <input checked="" type="checkbox"/> RPL	DevOps Foundations: Continuous Integration and Continuous Delivery Getting Started with Jenkins Travis CI
18.0 Understand and demonstrate continuous delivery	18.1 Articulate the value of continuous delivery in a development environment - Course 1, DevOps Foundations: Continuous Integration and Continuous Delivery Module 4 18.2 Understand CD principles through short, frequent, automated delivery of code - Course 1 DevOps Foundations: Continuous Integration and Continuous Delivery, Module 5 18.3 Demonstrate proficient use of continuous delivery tools - Course 2 Getting Started with Jenkins and Course 3 Travis CI	<input checked="" type="checkbox"/> O <input checked="" type="checkbox"/> Q&A <input checked="" type="checkbox"/> P <input type="checkbox"/> RA <input type="checkbox"/> S <input type="checkbox"/> PD <input type="checkbox"/> A <input checked="" type="checkbox"/> MT <input type="checkbox"/> EW <input checked="" type="checkbox"/> RPL	DevOps Foundations: Continuous Integration and Continuous Delivery Getting Started with Jenkins Travis CI

<p>19.0 Understand and demonstrate feature decoupling</p>	<p>19.1 Articulate the value of feature decoupling in a development environment 19.2 Demonstrate ability to proficiently use feature toggles 19.3 Understand knowledge of microservice architecture</p>	<p><input checked="" type="checkbox"/> O <input checked="" type="checkbox"/> Q&A <input checked="" type="checkbox"/> P <input type="checkbox"/> RA <input type="checkbox"/> S <input type="checkbox"/> PD <input type="checkbox"/> A <input checked="" type="checkbox"/> MT <input type="checkbox"/> EW <input checked="" type="checkbox"/> RPL</p>	<p>Feature Toggles</p>
<p>20.0 Understand and demonstrate DevOps automation</p>	<p>20.1 Articulate the value of automation to the development lifecycle 20.2 Demonstrate knowledge of the tools required to execute automation in a DevOps development team 20.3 Demonstrate use of automated run-books to minimize call-out for service failures 20.4 Demonstrate proficient use of UCD 20.5 Demonstrate proficient use of Ansible and scripted automation 20.6 Demonstrate proficient use of Chef</p>	<p><input checked="" type="checkbox"/> O <input checked="" type="checkbox"/> Q&A <input checked="" type="checkbox"/> P <input type="checkbox"/> RA <input type="checkbox"/> S <input type="checkbox"/> PD <input checked="" type="checkbox"/> A <input type="checkbox"/> MT <input type="checkbox"/> EW <input type="checkbox"/> RPL</p>	<p>DevOps: The Big Picture DevOps Foundations: Planning and Implementing a DevOps Strategy</p>
<p>21.0 Understand and demonstrate DevOps metrics and measurements</p>	<p>21.1 Understand the use and articulate the value of instrumentation and automated measurements to enable continuous improvement 21.2 Understand the various methodologies used to measure and track progress</p>	<p><input checked="" type="checkbox"/> O <input checked="" type="checkbox"/> Q&A <input checked="" type="checkbox"/> P <input type="checkbox"/> RA <input type="checkbox"/> S <input type="checkbox"/> PD <input type="checkbox"/> A <input checked="" type="checkbox"/> MT <input type="checkbox"/> EW <input checked="" type="checkbox"/> RPL</p>	<p>DevOps: The Big Picture DevOps Foundations: Planning and Implementing a DevOps Strategy</p>
<p>22.0 Understand and demonstrate knowledge of cloud computing fundamentals, including the various tools, services, and principles</p>	<p>22.1 Articulate the value of cloud infrastructures as it relates to efficiency, scalability, and resiliency of services 22.2 Articulate the value and use of containers in cloud computing 22.3 Demonstrate proficient use of containers like Docker and Vagrant 22.4 Articulate the value of platforms like OpenStack and IaaS 22.5 Demonstrate proficient use of cloud platforms 22.6 Articulate the value of container management systems 22.7 Demonstrate proficient use of service development practices, including adherence to 12-factor app rules</p>	<p><input checked="" type="checkbox"/> O <input checked="" type="checkbox"/> Q&A <input checked="" type="checkbox"/> P <input type="checkbox"/> RA <input type="checkbox"/> S <input type="checkbox"/> PD <input checked="" type="checkbox"/> A <input checked="" type="checkbox"/> MT <input type="checkbox"/> EW <input checked="" type="checkbox"/> RPL</p>	<p>IBM Cloud Essentials IBM Cloudant Docker and Kubernetes: The Big Picture Introduction to Containers, Kubernetes and OpenShift V2 Introduction to OpenStack</p>

<p>23.0 Understand and demonstrate the design patterns and practices for building cloud native services</p>	<p>23.1 Articulate the design patterns and practices for building highly scalable, resilient services and their lifecycles</p> <p>23.2 Demonstrate understanding of Kubernetes</p> <p>23.3 Demonstrate understanding of microservices architecture, including platforms like Liber8 and Istio</p> <p>23.4 Understand proficient understanding of circuit-breaker patterns like Hystrix and Istio</p>	<p><input checked="" type="checkbox"/> O</p> <p><input checked="" type="checkbox"/> Q&A</p> <p><input checked="" type="checkbox"/> P</p> <p><input type="checkbox"/> RA</p> <p><input type="checkbox"/> S</p> <p><input type="checkbox"/> PD</p> <p><input checked="" type="checkbox"/> A</p> <p><input checked="" type="checkbox"/> MT</p> <p><input type="checkbox"/> EW</p> <p><input checked="" type="checkbox"/> RPL</p>	<p>Get Started with Microservices Istio and IBM Cloud Kubernetes Service</p> <p>Beyond Basics Istio and IBM Cloud Kubernetes Service</p>
<p>24.0 Understand the relationship between scaling techniques, how to exploit them, and where</p>	<p>24.1 Articulate the relationship between scaling techniques, how to exploit them, and where</p> <p>24.2 Articulate the value of scale-out vs. scale-up</p> <p>24.3 Demonstrate use of scale-out vs. scale-up</p> <p>24.4 Understand the value of fail fast / recovery</p> <p>24.5 Understand the use of fail fast / recovery</p>	<p><input checked="" type="checkbox"/> O</p> <p><input checked="" type="checkbox"/> Q&A</p> <p><input type="checkbox"/> P</p> <p><input type="checkbox"/> RA</p> <p><input type="checkbox"/> S</p> <p><input type="checkbox"/> PD</p> <p><input type="checkbox"/> A</p> <p><input checked="" type="checkbox"/> MT</p> <p><input type="checkbox"/> EW</p> <p><input checked="" type="checkbox"/> RPL</p>	<p>OJT</p>
<p>25.0 Understand the various platforms, their differences, relative strengths / weaknesses, and integrations</p>	<p>25.1 Outline the various platforms, their differences, relative strengths / weaknesses, and integrations</p> <p>25.2 Demonstrate basic knowledge of z/OS</p> <p>25.3 Demonstrate basic knowledge of Linux</p> <p>25.4 Demonstrate basic knowledge of Mobile</p> <p>25.5 Demonstrate basic knowledge of IoT</p>	<p><input checked="" type="checkbox"/> O</p> <p><input checked="" type="checkbox"/> Q&A</p> <p><input checked="" type="checkbox"/> P</p> <p><input type="checkbox"/> RA</p> <p><input type="checkbox"/> S</p> <p><input type="checkbox"/> PD</p> <p><input checked="" type="checkbox"/> A</p> <p><input checked="" type="checkbox"/> MT</p> <p><input type="checkbox"/> EW</p> <p><input checked="" type="checkbox"/> RPL</p>	<p>z/OS Introduction - Redbook Video Course</p>
<p>26.0 Understand the topologies of enterprise solutions, and how clients use our portfolio of products and services together</p>	<p>26.1 Articulate the topologies of enterprise solutions, and how clients use our portfolio of products and services together</p> <p>26.2 Articulate the common use cases for APIs</p> <p>26.3 Successfully design, publish, and manage APIs</p> <p>26.4 Demonstrate understanding of z/OS Connect & use of APIs to expose data</p>	<p><input checked="" type="checkbox"/> O</p> <p><input checked="" type="checkbox"/> Q&A</p> <p><input checked="" type="checkbox"/> P</p> <p><input type="checkbox"/> RA</p> <p><input type="checkbox"/> S</p> <p><input type="checkbox"/> PD</p> <p><input type="checkbox"/> A</p> <p><input checked="" type="checkbox"/> MT</p> <p><input type="checkbox"/> EW</p> <p><input checked="" type="checkbox"/> RPL</p>	<p>Building an API Bridge to Your IBM z Systems Applications and Data</p> <p>z/OS Connect EE - IBM z/OS Connect Enterprise Edition v3</p>
<p>27.0 Understand and demonstrate the use of logging and</p>	<p>27.1 Articulate the value of logging and monitoring in an operational environment</p>	<p><input checked="" type="checkbox"/> O</p> <p><input checked="" type="checkbox"/> Q&A</p> <p><input checked="" type="checkbox"/> P</p>	<p>z/OS Introduction -</p>

monitoring tools and infrastructure	27.2 Demonstrate proficient use of logging and monitoring tools and infrastructure 27.3 Demonstrate ability to implement and/or comply with logging policies 27.4 Demonstrate ability to use dashboards for monitoring 27.5 Demonstrate understanding of how data traverses	<input type="checkbox"/> RA <input type="checkbox"/> S <input type="checkbox"/> PD <input type="checkbox"/> A <input checked="" type="checkbox"/> MT <input type="checkbox"/> EW <input checked="" type="checkbox"/> RPL	Redbook Video Course
28.0 Understand the need for mobility and migration of data from on-premise to cloud solutions, and the implications	28.1 Articulate the value in migrating data from on-premise to cloud 28.2 Articulate the challenges in migrating data from on-premise to cloud 28.3 Outline the process required to migrate data 28.4 Demonstrate understanding of how to work with system of record data on Z and expose it	<input checked="" type="checkbox"/> O <input checked="" type="checkbox"/> Q&A <input checked="" type="checkbox"/> P <input type="checkbox"/> RA <input type="checkbox"/> S <input type="checkbox"/> PD <input type="checkbox"/> A <input checked="" type="checkbox"/> MT <input type="checkbox"/> EW <input checked="" type="checkbox"/> RPL	OJT
29.0 Demonstrate basic understanding of Mainframe terminology and basic concepts	29.1 Demonstrate ability to describe and compare various IBM Z components: 29.1.1 Frame layout and cage usage 29.1.2 Server models, books, memory, and cache structure 29.1.3 Performance and millions of service units (MSUs)	<input checked="" type="checkbox"/> O <input checked="" type="checkbox"/> Q&A <input type="checkbox"/> P <input checked="" type="checkbox"/> RA <input type="checkbox"/> S <input checked="" type="checkbox"/> PD <input checked="" type="checkbox"/> A <input checked="" type="checkbox"/> MT <input checked="" type="checkbox"/> EW <input type="checkbox"/> RPL	Introduction to IBM z/OS Introduction to z/OS Commands and Panels Introduction to Systems Programming on IBM Z
30.0 Demonstrate basic understanding of the hierarchical and relational DBMS	30.1 Demonstrate ability to articulate how databases are used in a typical online business 30.2 Demonstrate ability to describe models for network connectivity for large systems 30.3 Demonstrate ability to explain the role of Db2 in online transaction processing 30.4 Demonstrate ability to list common Db2 data structures 30.5 Demonstrate ability to describe how SQL works on z/OS 30.6 Demonstrate ability to give an overview of application programming with Db2 30.7 Demonstrate ability to explain what the IMS components are 30.8 Demonstrate ability to describe the structure of the IMS Db subsystem	<input checked="" type="checkbox"/> O <input checked="" type="checkbox"/> Q&A <input type="checkbox"/> P <input checked="" type="checkbox"/> RA <input type="checkbox"/> S <input checked="" type="checkbox"/> PD <input checked="" type="checkbox"/> A <input checked="" type="checkbox"/> MT <input checked="" type="checkbox"/> EW <input type="checkbox"/> RPL	Db2 - Db2 Fundamentals v12 Db2 - Managing Db2 Operations v12 IMS Fundamentals
31.0 Demonstrate basic understanding of the concepts, functions and facilities of CICS	31.1 Demonstrate ability to describe the role of large systems in a typical online business 31.2 Demonstrate ability to list the attributes common to most transaction systems	<input checked="" type="checkbox"/> O <input checked="" type="checkbox"/> Q&A <input type="checkbox"/> P	IBM CICS Video Course Series

	<p>31.3 Demonstrate ability to explain the role of CICS in online transaction processing</p> <p>31.4 Demonstrate ability to describe CICS programs, CICS transactions, and CICS tasks</p> <p>31.5 Demonstrate ability to describe the CICS IMS components</p>	<input checked="" type="checkbox"/> RA <input type="checkbox"/> S <input type="checkbox"/> PD <input checked="" type="checkbox"/> A <input checked="" type="checkbox"/> MT <input checked="" type="checkbox"/> EW <input type="checkbox"/> RPL	<p>Modernize Applications with IBM CICS</p> <p>CICS TS: CICS Transaction Server Introduction 5.6</p> <p>CICS TS: CICS Explorer Fundamentals 5.6</p> <p>CICS TS: CICS Command Simulation 5.6</p>
--	---	--	---

Outline of Related Instruction

Professional Foundational Instruction is required for all apprentices.

Professional Foundational Instruction

Professional Foundational Instruction	Formal Training Hours
Employee Onboarding and Integration Success Skills	50

Technical Foundational Instruction

Technical Foundational Instruction	Formal Training Hours
Principal and Practices <ul style="list-style-type: none"> • Agile practices, including operations, program fundamentals, and project and change management • Leadership skills • Big Data and Analytics 	27
Software Engineering Fundamentals <ul style="list-style-type: none"> • Web development fundamentals • Clean coding and social coding behaviors and practices • Introductions to functional and imperative languages • COBOL programming • Test Driven Development 	122
DevOps Fundamentals <ul style="list-style-type: none"> • Continuous integration and continuous delivery practices • Feature Decoupling • DevOps tools, including Chef, Travis, Ansible, and Jenkins 	11
Platforms, Services, and Solutions <ul style="list-style-type: none"> • Cloud computing fundamentals • Enterprise solutions • IBM z/OS Practitioner Course 	131
Total	341