

Digital Industries Apprenticeships
Software Developer: Occupational Brief
June 2016



Software Developer: Competencies

The Competency Standards	Definition of the Minimum Requirement	Examples and illustrations – NOT the curriculum
<p>Logic: writes good quality code (logic) with sound syntax in at least one language</p>	<p>Apprentices can write code to achieve the desired functionality and which is easy to read and understand, with good naming, indentation and commenting, and applying the fundamentals of good coding</p> <ul style="list-style-type: none"> • development paradigms (where this is object oriented programming this must include inheritance, abstractions, encapsulation, polymorphism) • software programming languages • software development tools (IDEs) • writing programs and methods • language-specific idioms • logic and flow-of-control <p>Apprentices can apply</p> <ul style="list-style-type: none"> • Elements of programming – variables, assignment statements, data types, conditionals, loops, arrays, and input/output. • Functions - modular programming dividing a program into components that can be independently debugged, maintained, and reused writing at least two reusable functions • Algorithms and data structures - classical algorithms for sorting and searching, and fundamental data structures. 	<p>A wide range of software development tools including</p> <ul style="list-style-type: none"> • Integrated Development Environments (IDE's) • Version control systems • Configuration management tools <p>For example in Java the ability to code variables assignment statements data types conditionals statements loops arrays</p> <p>For example in Java to be able to declare and invoke methods correctly demonstrate parameter passing and returning values overloading and overriding</p> <p>For example in Java demonstrate the use of collection classes the difference between standard arrays and collection classes</p>

The Competency Standards	Definition of the Minimum Requirement	Examples and illustrations – NOT the curriculum
<p>User interface: can develop effective user interfaces for at least one channel</p>	<p>Apprentices can apply the fundamental concepts of human-computer interaction or user experience design, the development practices leading to a high-quality user interface, and the programming techniques required to construct a graphical user interface.</p> <p>Can interact with screen or UI designers to ensure the logic layer integrates with the user interface</p> <p>Can develop user interface coding and implementation - techniques for building user interfaces – for at least one channel</p> <p>Can interact with testers to optimise the user interface</p>	<ul style="list-style-type: none"> <input type="checkbox"/> UI prototyping tools <input type="checkbox"/> UI design tools <p>For example in Java be able to design a GUI using</p> <ul style="list-style-type: none"> Swing/Java FX. Components Containers Layout Managers
<p>Data: can effectively link code to the database/data sets</p>	<p>Apprentices can link to a range of database types and embed data queries within their code</p> <ul style="list-style-type: none"> Can make a connection to a database Can execute CRUD statements on the database Can use one-off queries and stored procedures Can transform returned data in to format the application requires 	<ul style="list-style-type: none"> • Databases • SQL <p>For example, in Java use JDBC to create connections</p> <ul style="list-style-type: none"> statements result sets
<p>Test: can test code and analyse results to correct errors found using either V-model manual testing and/or using unit testing</p>	<p>Apprentices can test and analyse their code to identify errors as soon as possible in the coding process and on an interactive basis</p> <ul style="list-style-type: none"> Can apply test and debugging strategies Can design and develop manual or unit tests Can test code segment functionality against requirements Can assess test results against expected results and acceptance criteria 	<ul style="list-style-type: none"> <input type="checkbox"/> Debugging tools <input type="checkbox"/> Unit test tools (e.g. Junit etc) <p>For example in Java to write Junit test cases to</p> <ul style="list-style-type: none"> test for individual methods test all routes through an application. check expected results against actual results
<p>Problem solving: can apply structured techniques to problem solving, can debug code and can understand the structure of programmes in order to</p>	<p>Can use a minimum of two problem solving tools and techniques to identify and resolve programming issues</p> <ul style="list-style-type: none"> Can apply structured problem solving methods 	<ul style="list-style-type: none"> • Problem solving tools and techniques

The Competency Standards	Definition of the Minimum Requirement	Examples and illustrations – NOT the curriculum
identify and resolve issues	Can apply problem-solving techniques to programming activities	For example, in Java using debugger.
Design: can create simple data models and software designs to effectively communicate understanding of the program, following best practices and standards	<p>Can take a high level design and can interpret and convert the design in to simple data models and/or programme modules to communicate it to others</p> <p>Can apply a software design methodologies (e.g., structured or object-oriented)</p> <p>Can use standard design notation such as UML</p> <p>Can apply data modelling</p> <p>Can apply reconcile design against analysis models</p> <p>Can design software solutions to meet requirements</p>	<ul style="list-style-type: none"> • Software design tools • Data modelling tools
Analysis: can understand and create basic analysis artefacts, such as user cases and/or user stories	<p>Can take a variety of data and business requirements and convert them in to basic analysis artefacts to understand and can clarify the intended use of the proposed software</p> <p>Can identify and represent required functionality (e.g. use cases)</p> <p>Can identify and represent activity workflow (e.g. activity diagrams)</p>	<ul style="list-style-type: none"> • Software analysis tools • Case development tools • Activity diagram tools
Deployment : can understand and utilise skills to build, manage and deploy code into enterprise environments	<p>Can package and build completed programmes as appropriate to the resources available for deployment and for migration to different environments, including</p> <p>developing appropriate user documentation</p> <p>planning for user training and data migration.</p>	<ul style="list-style-type: none"> • Software deployment tools
Development lifecycle: can operate at all stages of the software development lifecycle, with increasing breadth and depth over time with initial focus on build and test.	Apprentices can operate as software developers showing a good understanding of the other phases of the software development lifecycle and the deliverables that are produced at each stage and as relevant to the development methodology (waterfall, agile, test led etc)	<ul style="list-style-type: none"> • Software development lifecycle methods and tools (e.g. RUP)

The Competency Standards	Definition of the Minimum Requirement	Examples and illustrations – NOT the curriculum
	<p>In addition to the stages above, the apprentice can also operate in the support and maintenance phases</p> <p>Can advise third line support for relevant applications Can fix bugs and deal with change requests</p>	
<p>Can apply good practice approaches according to the relevant paradigm (for example object oriented, event driven or procedural)</p>	<p>Apprentices can identify and follow standards and good practice that can improve programming efficiency, style and quality, including</p> <p>programming standards, both organisational and external generic best practices including readability, reusability, maintainability best practice approaches of different paradigms and languages</p>	<ul style="list-style-type: none"> • In company standards for software coding • External standards for software coding • Best practice frameworks and standards
<p>Can interpret and follow software designs, functional/technical specifications, company defined 'coding standards' or industry good practice for coding, testing frameworks and methodologies, and company, team or client approaches to continuous integration, version and source control</p>	<p>Apprentices can adapt to the employers domain and context for software development and interpret and follow the software development approach being implemented</p> <p>Can read software designs and functional/technical specifications, especially those based upon the employer domain and context Can identify, interpret and follow 'coding standards' Can identify, interpret and follow best practice coding approaches for specific paradigms and languages Can identify, interpret and follow company, team or client approaches to continuous integration, version and source control</p>	<ul style="list-style-type: none"> • Typically in-company documentation
<p>Can respond to the business environment and business issues related to software development</p>	<p>Apprentices can apply the following considerations when working on projects</p> <p>Business context Business drivers (efficiency gains, increased functionality and improved quality of outputs)</p>	

The Competency Standards	Definition of the Minimum Requirement	Examples and illustrations – NOT the curriculum
Can operate effectively in their own businesses, their customers' and the industry's environments	Can demonstrate working within operational requirements such as health and safety, budgets, brands and normal business protocols	
Can apply the maths required to be a software developer (eg algorithms, logic and data structures)		

Software Developer: Knowledge and Understanding

Knowledge Module 1: Software Development Methodologies (for Level 4 Software Developer Apprenticeship)

The Knowledge Standards	Definition of the Minimum Requirements
Understands and operates at all stages of the software development lifecycle	a) What is the software development lifecycle (SDLC) b) What are the seven generic stages and their high-level deliverables from each stage c) What are the main activities in each stage
Understands the similarities and differences (taking into account positives and negatives of both approaches) between agile and waterfall software development methodologies	a) What is the agile development method b) What is the waterfall development method c) What are the strengths and weaknesses of both approaches
Understands how teams work effectively to produce software and contributes appropriately	a) What are the roles that need to be undertaken b) How are these roles distributed across a team c) What team-working aspects are needed to ensure effective delivery of projects

The following Vendor or Professional Certifications exempt apprentices from this knowledge module

BCS Systems Development Essentials

Knowledge Module 2: Software Languages (for Level 4 Software Developer Apprenticeship)

The Knowledge Standards	Definition of the Minimum Requirements
Understands and applies software design approaches and patterns and	a) Software design approaches

can interpret and implement a given design, compliant with security and maintainability requirements	<ul style="list-style-type: none"> b) Software patterns c) Documenting software designs d) Secure development e) Designing for software maintainability and re-use
Understands and applies the maths required to be a software developer (e.g. algorithms, logic and data structures)	<ul style="list-style-type: none"> a. Understanding of basic algorithmic processing to define the problem and/or solution b. Elements of programming logic - variables; assignment statements; data types; conditionals; loops; arrays; and input/output, Knowledge of at least two data structures – such as Arrays or Collection Classes

The following Vendor or Professional certifications exempt apprentices from this knowledge module

Cloud certified developer apache

Hadoop

C++

PHP

Drupal

Oracle SQL Developer

Oracle Java Certified

MCP.net

MTA / MCP programming in HTML5 with Javascript and CSS3

C#

Software Developer: Grading – Minimum Criteria

Minimum Requirements	Attributes to be explored (illustrative)	Assessment techniques (illustrative)
<i>The What: what the apprentice has shown they can do</i>		
The minimum requirements are set out in the occupational brief		
<i>The How: the way in which the work has been done</i>		
<p>Apprentices can demonstrate the full range of skills, knowledge and behaviours required to fulfil their job role</p> <p>Apprentices can demonstrate how they contribute to the wider business objectives and show an understanding of the wider business environments</p>	<p>a. The job:</p> <ul style="list-style-type: none"> • Understanding the scope of the job role • Knowing what skills, knowledge and behaviours are needed to do the job well • Being aware of their own strengths in the job role, and any areas for improvement • Appreciating who else is important, for them to do their job and fulfil the role effectively (e.g. colleagues, managers, other stakeholders) • Being aware of potential risks in the job role (e.g. security, privacy, regulatory) • Using personal attributes effectively in the role, e.g. entrepreneurship • Understanding how the job fits into the organisation as a whole • Knowing what the next steps in their career might be <p>b. The organisation:</p> <ul style="list-style-type: none"> • Understanding the goals, vision and values of the organisation • Knowing how they contribute to these in their own work • Being aware of the commercial objectives of the tasks/ projects they are working on 	<p>Factual questioning, e.g.:</p> <ul style="list-style-type: none"> • What are the main risks in your work? • How does the regulatory environment affect what you do? • What are the organisation's commercial goals? • What factors are shaping your industry at the moment? <p>Process questioning, e.g. :</p> <ul style="list-style-type: none"> • What are the key things that make a difference to how well you do your job? • How do other people you work with, including customers and clients, affect the way you work?

Minimum Requirements	Attributes to be explored (illustrative)	Assessment techniques (illustrative)
	<ul style="list-style-type: none"> • Understanding the importance of meeting or exceeding customers' requirements and expectations • Being in tune with the organisation's culture • Knowing how the organisation works, including its informal culture, internal networks etc. <p>c. The environment:</p> <ul style="list-style-type: none"> • Being aware of the position of the organisation in the economy and its contribution to society • Understanding the key external factors that shape the way the organisation function, e.g. regulation • Knowing how the organisation can gain advantage in the industry, e.g. through innovation, technology, customer service etc. 	
<p>Apprentices can demonstrate the ability to use both logical and creative thinking skills when undertaking work tasks, recognising and applying techniques from both.</p> <p>Apprentices can show that they recognise problems inherent in, or emerging during, work tasks, and can tackle them effectively.</p>	<p>a. Logical thinking:</p> <ul style="list-style-type: none"> • Understanding initial premise(s) and preconditions • Analysing situations from known facts • Recognising the conclusion to be reached • Proceeding by rational steps • Evaluating information, judging its relevance and value • Supporting conclusions, using reasoned arguments and evidence <p>a. Creative thinking:</p> <ul style="list-style-type: none"> • Looking at situations from a fresh perspective • Exploring ideas and possibilities • Making connections between different aspects • Questioning assumptions • Generating solutions that may be imaginative or unconventional • Devising new approaches • Adapting ideas and approaches as conditions or circumstances change 	<p>Behavioural questioning, e.g. :</p> <ul style="list-style-type: none"> • Tell me about how you approached this task • How did you recognise the problem? • Give me an example of thinking differently about a task. • Describe a decision that you made using a logical approach.

Minimum Requirements	Attributes to be explored (illustrative)	Assessment techniques (illustrative)
	<p><i>b. Problem-solving:</i></p> <ul style="list-style-type: none"> • Analysing situations • Defining goals • Developing solutions • Prioritising actions • Dealing with unexpected occurrences 	
<i>The With Whom – the personal and interpersonal qualities the apprentice has brought to internal and external relationships</i>		
<p>Apprentices can manage relationships with work colleagues, including those in more senior roles, customers/clients and other stakeholders, internal or external and as appropriate to their roles, so as to gain their confidence, keep them involved and maintain their support for the task/project in hand.</p> <p>Apprentices can establish and maintain productive working relationships, and can use a range of different techniques for doing so.</p>	<p>a. Managing relationships:</p> <ul style="list-style-type: none"> • Understanding the value and importance of good relationships • Adopting a way of working that maintains and improves relationships • Involving other people in decisions and actions • Influencing others by listening to and incorporating their ideas and views • Acknowledging other people's accomplishments and strengths • Overcoming barriers that prevent productive action • Managing conflict constructively • Promoting teamwork by encouraging others to participate <p>b. Customer/client relationships:</p> <ul style="list-style-type: none"> • Establishing contact with customers/clients and keeping in touch • Keeping customers/clients informed ('no surprises') • Understanding their requirements, including constraints and limiting factors • Setting reasonable expectations • Involving them in decisions and actions ('co-production') • Interacting positively with them 	<p>Process questioning, e.g. :</p> <ul style="list-style-type: none"> • What are the most important ways that you have found to manage relationships in your work? • Who do you manage relationships with? • What do customers and clients respond to well? • In what ways do you try to make relationships with customers and clients more productive? • What are the main differences between customers/clients and other stakeholders? • How do you manage the various stakeholders?

Minimum Requirements	Attributes to be explored (illustrative)	Assessment techniques (illustrative)
	<ul style="list-style-type: none"> • Communicating in different ways • Providing a complete answer in response to queries ('transparency', 'full disclosure') • Actively seeking feedback <p>b. Stakeholders:</p> <ul style="list-style-type: none"> • Understanding who they are and what their 'stake' is • Prioritising stakeholders in terms of their importance, power to affect the task and interest in it • Using stakeholders' views to shape projects early on • Gaining support from stakeholders, e.g. to win resources • Agreeing objectives • Managing expectations 	
<p>Apprentices can communicate effectively with a range of people at work, one-to-one and in groups, in different situations and using a variety of methods.</p> <p>Apprentices can demonstrate various methods of communication, with an understanding of the strengths, weaknesses and limitations of these, the factors that may disrupt it, and the importance of checking other people's understanding.</p>	<p>a. Intention/purpose:</p> <ul style="list-style-type: none"> • Understanding the purpose of communicating in a particular situation or circumstance (e.g. inform, instruct, suggest, discuss, negotiate etc.) • Checking that the person/people with whom one is communicating also understand the purpose • Being sensitive to the dynamics of the situation • Being aware of anything that might disrupt the effectiveness of the communication (e.g. status, past history) <p>b. Method:</p> <ul style="list-style-type: none"> • Knowing the range of possible communication methods (e.g. spoken, written, graphical, multimedia) • Choosing a good, appropriate method for the situation • Being aware of the limitations of the chosen method, and the possible risks of miscommunication (e.g. ambiguity) • Taking account of the affective dimensions of the method (e.g. body language, tone of voice, eye contact, facial expression etc.) <p>c. Execution:</p>	<p>Observation, e.g.:</p> <ul style="list-style-type: none"> • Examples of written, graphical and multimedia communication in the summative portfolio and synoptic project • How the apprentice communicates verbally and non-verbally during the course of the interview, including listening and responding <p>Funnelling questioning, e.g. :</p> <ul style="list-style-type: none"> • Tell me about a time when you needed to communicate something important, e.g. to <ul style="list-style-type: none"> • Colleagues • Superiors • Customers/clients • What worked well in this situation? • What didn't work as well as you had hoped? • What did you learn about communication from this situation (or any other)?

Minimum Requirements	Attributes to be explored (illustrative)	Assessment techniques (illustrative)
	<ul style="list-style-type: none">• Expressing oneself clearly and succinctly, but not over-simplifying• Checking that the other person/people understand what is being expressed• Taking account of the potential barriers to understanding (e.g. filtering, selective perception, information overload)• Modifying the purpose and methods of communication during a situation in response to cues from the other person/people	