

9628-02 Level 4 Diploma in Software Language (for the Level 4 Software Developer Apprenticeship)

9628-402 Software Language

Sample question paper answer sheet

Pass mark 25/36 (70%)

Question	Answer Key	Test specification reference
1	A	1.1a Describe the software design phase in the software design lifecycle
2	D	1.1c Describe the features of <ul style="list-style-type: none"> • A structured design approach • An object oriented design (OO) approach
3	B	1.1d Explain why different design approaches are used
4	A	1.1e Explain the relative importance of design considerations and the resulting trade-offs that are made
5	C	1.1f Explain the role of performance modelling and simulation
6	B	1.1g Explain how to utilise design patterns
7	C	1.2a Explain the benefits of encapsulation and modularisation for a given design
8	A	1.2b Explain the benefits of abstraction for a given design
9	D	1.2c Explain the benefits of coupling and cohesion for a given design
10	B	1.2d Explain the benefits of component reuse for a given design
11	C	1.2e Explain the benefits of using standard patterns for a given design
12	D	1.3a Describe the advantages and disadvantages of a structured programming paradigm

13	A	1.3b Describe the advantages and disadvantages of an object oriented programming paradigm
14	B	1.3c Explain when it is appropriate to use specific programming paradigms
15	B	1.3d Explain how a structured programming paradigm supports code reuse
16	D	1.3e Explain how an object oriented programming paradigm supports code reuse
17	A	1.3f Contrast the maintainability of code from different paradigms
18	A	2.1a Explain the importance of considering security as part of the software development process
19	D	2.1d Explain the principle of least privilege in software development
20	A	2.1e Explain the advantages of reusing previously tested code in the context of security
21	D	2.1f Describe how security threats are mitigated when developing code
22	B	2.1f Describe how security threats are mitigated when developing code
23	C	2.2a Describe how maintenance requirements should be considered in the software development lifecycle
24	A	2.2b Describe activities in the maintenance phase of the software development lifecycle
25	A	2.2c Explain the need for comprehensive software documentation for software maintenance
26	C	2.2c Explain the need for comprehensive software documentation for software maintenance
27	B	2.2d Explain the importance of complying with standards for the maintenance of software
28	B	3.1a Describe the concept of computational problems (Recognise/solve common computational problems)
29	B	3.1b Describe the concept of algorithms (Recognise common uses of algorithms)
30	D	3.2a Explain the use of tools and techniques for creating abstractions that represent a problem

31	B	3.2b Recognise algorithms that solve a problem efficiently
32	D	3.2c Describe the use of sets, functions, logic and proofs in algorithm design
33	B	3.2d Explain how to use formal mathematical notation to present algorithms
34	D	3.3a Describe a range of standard data structures
35	A	3.3b Describe common algorithms for sorting, searching and optimisation
36	B	3.3b Describe common algorithms for sorting, searching and optimisation