

Topic 12: Software Development

EAL keyword sheet - Cambridge International AS & A Level Computer Science 9618. Write the word in your own language in the last column.

Keyword	Meaning	Example sentence	In my language
software development life cycle (SDLC)	The stages of making software.	<i>The SDLC includes analysis, design and testing.</i>	
analysis	Finding out what the user needs.	<i>Analysis includes interviews and questionnaires.</i>	
requirements specification	A document that lists what the system must do.	<i>The requirements specification was signed.</i>	
design	Planning how the system will work.	<i>Design uses flowcharts and structure charts.</i>	
coding / development	Writing the program.	<i>Coding takes place after design.</i>	
testing	Checking that the program works correctly.	<i>Testing found three errors.</i>	
implementation / deployment	Putting the new system into use.	<i>Deployment happens after successful testing.</i>	
maintenance	Fixing problems and improving the system after it is in use.	<i>Maintenance keeps the software up to date.</i>	
waterfall model	A life cycle where each stage is completed before the next one starts.	<i>The waterfall model is easy to manage.</i>	
iterative model	A life cycle where the software is built and improved in repeated cycles.	<i>An iterative model allows changes.</i>	
rapid application development (RAD)	Building quick prototypes and changing them with user feedback.	<i>RAD gives fast results.</i>	
prototyping	Making an early version of the software to test ideas.	<i>Users tested the prototype.</i>	
structure chart	A diagram showing the modules of a program and how they connect.	<i>The structure chart shows three modules.</i>	
state-transition diagram	A diagram that shows states and the events that change them.	<i>A state-transition diagram shows a lock.</i>	
test plan	A list of tests, test data and expected results.	<i>The test plan has twelve tests.</i>	

Keyword	Meaning	Example sentence	In my language
test data	Values used to test a program.	<i>Test data includes normal and abnormal values.</i>	
normal / abnormal / boundary data	Valid data / invalid data / values at the limits.	<i>Boundary data tests the edges of a range.</i>	
black-box testing	Testing the inputs and outputs without looking at the code.	<i>Black-box testing checks behaviour.</i>	
white-box testing	Testing with knowledge of the code.	<i>White-box testing checks every path.</i>	
dry run	Checking a program by hand.	<i>A dry run found the error.</i>	
integration testing	Testing that modules work together.	<i>Integration testing follows module testing.</i>	
alpha / beta testing	Testing by developers / testing by selected users before release.	<i>Beta testing finds real-world problems.</i>	
acceptance testing	Testing by the customer to check it meets their needs.	<i>The client carried out acceptance testing.</i>	
stub	A simple placeholder for a module that is not finished.	<i>A stub returns a fixed value.</i>	
documentation	Written information about the system: user guide and technical documents.	<i>The user documentation explains how to log in.</i>	
corrective / adaptive / perfective maintenance	Fixing errors / adjusting to new conditions / improving performance.	<i>Perfective maintenance makes the software faster.</i>	