

SDD PRELIMINARY CHANGES SUMMARY

This document aims to highlight recent changes made to the NSW Software Design and Development Preliminary syllabus. The original syllabus will be examined for the last time at the 2011 HSC and the revised syllabus will be examined from the 2012 HSC onwards. Hence, Year 11 Preliminary classes from 2011 onwards study the revised SDD syllabus.

This document aims to assist those schools intending to use the Second Edition of my SDD Preliminary text, however it may well be a useful summary for all teachers of SDD. Please feel free to contact me directly with any questions.

Regards,



Sam Davis
Parramatta Education Centre
E: sam.davis@pedc.com.au
P: 4632 7987
F: 4632 8002

BRIEF SUMMARY OF SDD PRELIMINARY COURSE CHANGES

8.1 Concepts and issues in the design and development of software (30%)

8.1.1 Social and ethical issues

Command line interface, email and social networking have been added as significant applications/design features. Ergonomic equipment design and placement has been removed, together with related overuse injuries. Ergonomics now deals solely with software design. Privacy of personal details has been added. Open source and creative commons licences are now included.

8.1.2 Hardware and software

Specific input, output and storage devices specified, however students are required to study just one of each type. Incremental compilation deleted. More detail on operating systems (and related utilities) is included.

8.1.3 Software development approaches

Phases in the software development cycle are now the same as the HSC course. Agile approach added. Combination of approaches explicitly stated.

8.2 Introduction to software development (50%)

8.2.1 Defining and understanding the problem, and planning and designing software solutions

All system models are now in the preliminary course including IPO diagrams, context diagrams, DFDs, structure charts, systems flowcharts and data dictionaries. Octal has been removed. Using sequential files for storing records is now explicitly in the preliminary course. New algorithm syntax for functions is specified in the Software and Course Specifications.

8.2.2 Implementing software solutions

BNF removed. Coding a login process is now required. Global variables and parameter passing now explicitly included. Distinction between and the use of procedures and functions explicitly included. Storyboards now included in preliminary course.

8.2.3 Testing and evaluating software solutions

Minor changes to emphasise desk checks and comparison of results with expected outputs.

8.2.4 Maintaining software solutions

Minor changes to emphasise internal documentation and copyright issues.

8.3 Developing software solutions (20%)

Project management has been moved to the start of the topic which better reflects its importance in regard to student projects. This is followed by a list of the system models and other documentation types that are now included in the preliminary course. The developing software solutions subtopic provides a good summary of the significant tasks performed during each phase of the software development cycle.

DETAILED SUMMARY OF CHANGES

Preliminary Course

8.1 Concepts and issues in the design and development of software (30%)

8.1.1 Social and ethical issues

Command line interface, email and social networking have been added as significant applications/design features. Ergonomic equipment design and placement has been removed, together with related overuse injuries. Ergonomics now deals solely with software design. Privacy of personal details has been added. Open source and creative commons licences are now included.

<i>Description of change</i>	<i>Old Syllabus deletions</i>		<i>New syllabus (Preliminary classes from 2011 onwards) additions</i>		<i>Davis SDD Prelim Text Second Edition</i>
New content added under evolution of software applications			Command line interface, email, social networking applications	p14	p4,6,11
Copyright explicitly included			Copyright	p14	p17+
Different software licence agreements listed in course specifications			Public domain, shareware, freeware, open source (GNU licence), site licence, creative commons	Course Specs	p19+
Shareware removed as a source of code	shareware	p15			
RSI and overuse injuries removed	effects of prolonged use of software, including RSI and injuries created by overuse	p14			
Procedures to prevent injuries removed	procedures to prevent and minimise injuries	p14			
Ergonomic furniture design and placement removed	ergonomically designed and placed equipment	p14			
Ergonomics of software design modified	acceptable response time in software	p14	effectiveness of screen design	p15	p25+
Social inclusivity deleted - presumably because other perspectives encompass social perspective	social perspectives	p15			
Privacy of individual's data/identity added			Privacy - need to protect an individual's data and identity	p15	p40
Technical skills added as a required skill			technical skills	p15	p43

8.1.2 Hardware and software

Specific input, output and storage devices specified, however students are required to study just one of each type. Incremental compilation deleted. More detail on operating systems (and related utilities) is included.

<i>Description of change</i>	<i>Old Syllabus deletions</i>		<i>New syllabus (Preliminary classes from 2011 onwards) additions</i>		<i>Davis SDD Prelim Text Second Edition</i>
Input devices explicitly specified in the course specifications. Students required to study one in detail.			Select ONE device from each: Input- keyboard, mouse, scanner, RFID, barcode reader, graphics, tablet, microphone	Course Specs	p52-64
Output devices explicitly specified in the course specifications. Students required to study one in detail. As CD and DVD are storage devices, my text deals with these under the storage heading.			Select ONE device from each: Output- laser printer, inkjet printer, CRT, LCD, plasma, CD, DVD, data projector, speakers	Course Specs	p65-79
Secondary storage devices explicitly specified in the course specifications. Students required to study one in detail. I also cover primary storage within my text.			Select ONE device from each: Storage- CD, DVD, flash drive, hard drive	Course Specs	p80-93
Detail of the functions performed by operating systems (and selected utilities) is now required.	System software, including utility software. Characteristics of different operating systems including command-based or GUI, multi-tasking	p16 p17	Operating systems and utilities, such as: file compression, defragging, virus checking, embedded licence installation counts. Functions of operating systems: provide interface to hardware, provide interface to user, provide interface to software applications, control the concurrent running of software applications, manage system resources: time (multitasking), memory, data, hardware devices. (Learn to) Identify typical tasks performed by operating systems: batch job scheduling, emulation.	p16 p17 Course Specs	p99-104
Incremental compilation removed.	incremental compilation	p17			
Initiation and running an application is more accurately (and simply) described.	The initiation and running of an application: start fetch-execute cycle, locate on disk, load into RAM, display start screen, wait for user input.	p17	The initiation and running of an application by the operating system: locate and load application, hand control to application, start fetch-execute cycle for application.	p17	p119
Event driven versus sequential approach removed, however I have left this in the textbook as no doubt most classes will code both types.	Event driven versus sequential approach	p17			p112-113

8.1.3 Software development approaches

Phases in the software development cycle are now the same as the HSC course. Agile approach added. Combination of approaches explicitly stated.

<i>Description of change</i>	<i>Old Syllabus deletions</i>		<i>New syllabus (Preliminary classes from 2011 onwards) additions</i>		<i>Davis SDD Prelim Text Second Edition</i>
Structured approach stages changed to be the same as the HSC course	Defining the problem, planning, building, checking and modifying.	p18	Defining and understanding the problem, planning and designing, implementing, testing and evaluating, maintaining.	p18	p124-128
Less emphasis on teams for structured approach and more emphasis on formal stages. Agile requires a multi-skilled team; hence it makes sense to deemphasise team for the structured approach.	team approach	p18	characteristics of structured approach - distinct formal stages	p18	p124
Agile approach added.			Agile approach (and all dot points)	p18	p128
Prototyping now focuses on limited (model) and evolutionary (progressive refinement) prototypes.	Characteristics of the prototyping approach, including: non-formal, shorter time period, small-scale projects, small budgets Involvement of personnel, including programmer and users Links with structured approach	p18 p19	Modelling of a proposed solution or part of a solution. Progressive refinement of the model in response to feedback. (Learn to) Design and develop a limited prototype as a proposed solution, or part of a solution, to a problem.	p18	p131
RAD approach more accurately described.	Coding languages used Relationship of programmer to end user Involvement of personnel, including developer and end user	p19	Use of existing routines Use of appropriate applications to develop the RAD solution: drag and drop programming environments, common application packages such as spreadsheets, databases Communication between developer and client	p19	p138
End user approach reworded to emphasise the user is the developer rather than the nature of the project.	Use of standard software packages, lack of formal stages, short time period, potential long-term/small-scale project, low budgets, end user is the developer	p19	end user as the developer and maintainer, typically uses RAD and/or prototyping, the developer is the client, therefore there are no communication issues, small budget and/or short time period for development	p19	p142
Combination of approaches is now emphasised			Software solutions that have been developed using a single approach, software solutions that have been developed using a combination of approaches	p19	Throughout Chapter 3.

8.2 Introduction to software development (50%)

8.2.1 Defining and understanding the problem, and planning and designing software solutions

All system models are now in the preliminary course including IPO diagrams, context diagrams, DFDs, structure charts, systems flowcharts and data dictionaries. Octal has been removed. Using sequential files for storing records is now explicitly in the preliminary course. New algorithm syntax for functions is specified in the Software and Course Specifications.

<i>Description of change</i>	<i>Old Syllabus deletions</i>		<i>New syllabus (Preliminary classes from 2011 onwards) additions</i>		<i>Davis SDD Prelim Text Second Edition</i>
Topic name changed to reflect HSC course	Defining the problem and planning software solutions	p20	Defining and understanding the problem, and planning and designing software solutions	p20	p149
Detail added to understanding the problem including specifying IPO diagrams			Clarification of the specifications, performance requirements, IPO diagrams.	p21	p150
Outline of syllabus meaning of system, program, module and subroutine added.			A system comprises all the programs in the suite, a program comprises all of the modules required to perform the required task, a module is a group of subroutines that together achieve a subtask, a subroutine is a set of statements that perform a logical task.	p21	p155
System models namely context diagrams, DFDs, structure charts and systems flowcharts are now in the preliminary course.			Document a proposed non-complex software solution: represent the flow of data through a system using a context diagram, represent a system using a data flow diagram (DFD) to show its components and the data transferred between them, represent a system using a structure chart to show the interrelationship between the component modules, represent a system using a systems flowchart to show its component modules, files and media.	p21	p158
Date and currency removed as a data type.	date and currency format	p21			
Integers in octal removed	Integer representation in binary , decimal, octal and hexadecimal	p21	Integer representation in binary , decimal and hexadecimal	p21	
Character representation including ASCII and binary, decimal, hexadecimal added			Characters represented as numbers in binary, decimal and hexadecimal. Interpret and use an ASCII table.	p21	p174
Records in sequential files explicitly included	Sequential files	p21	Use of records in sequential files	p21	p206
Data dictionary now in prelim course			Create a data dictionary which defines the data appropriately.	p21	p185
Subroutines added as a control structure and new RETURN syntax for algorithms			Use of subroutines (Includes new RETURN syntax for function algorithms -refer p54-56 of the Software and Course Specifications document)	p22	p200

<i>Description of change</i>	<i>Old Syllabus deletions</i>		<i>New syllabus (Preliminary classes from 2011 onwards) additions</i>		<i>Davis SDD Prelim Text Second Edition</i>
Software structure expanded and includes stubs.	Software structure: subroutines, modularity.	p21	Software structure: use of a clear uncluttered mainline and subroutines, use of a modular approach, use of stubs to represent incomplete modules	p22	p200 p244 (Ch 5) p319 (Ch 7)
Summing arrays of numbers specified			Add the contents of an array of numbers	p22	p206
Benefits of structured algorithms explicitly included.	Historical events that led to the development...	p21	Benefits of using structured algorithms: ease of development, ease of understanding, ease of modification	p22	Throughout Chapter 4
Converting source code to algorithms added			Represent code from different sources as an algorithm to assist in understanding its purpose and to assess its relevance in a proposed solution	p22	p323 (Ch 7)

8.2.2 Implementing software solutions

BNF removed. Coding a login process is now required. Global variables and parameter passing now explicitly included. Distinction between and the use of procedures and functions explicitly included. Storyboards now included in preliminary course.

<i>Description of change</i>	<i>Old Syllabus deletions</i>		<i>New syllabus (Preliminary classes from 2011 onwards) additions</i>		<i>Davis SDD Prelim Text Second Edition</i>
Topic name changed to reflect HSC course	Building software solutions	p22	Implementing software solutions	p23	p217
BNF removed completely	meta-languages including BNF, EBNF and railroad diagrams	p23	meta-languages including: EBNF, railroad diagrams	p24	
Use of manuals/help to determine syntax explicitly included			language syntax: specified through meta-languages in manuals and help documentation	p24	Throughout Chapter 5
Syntax for subroutines explicitly included			Use of subroutines or procedures	p24	Throughout Chapter 5
Boolean data type syntax added and sequential files removed as they are not classified as a data structure	The syntax of the statements used to define and use a range of data types, including: integer, string, floating point, one-dimensional array, record and sequential files	p23	The syntax used to define and use a range of data types and data structures, including integer, string, floating point/real, Boolean, one-dimensional array and records	p24	p227
Convert algorithms to source code now explicitly included.			Developing source code: converting algorithms into source code using syntactically correct statements	p24	Throughout Chapter 5
Stub usage more clearly described.	Stubs: used to check the connection between modules of code	p23	Stubs used to check the flow of execution or used to replace subroutines/modules during testing to check if that section of the code is the cause of an error	p24	p244

<i>Description of change</i>	<i>Old Syllabus deletions</i>		<i>New syllabus (Preliminary classes from 2011 onwards) additions</i>		<i>Davis SDD Prelim Text Second Edition</i>
Words to numbers deleted and replaced with login process.	Standard routines, such as data validation, date conversion and words to numbers.	p24	Standard logic, such as: a login process, data validation, conversion between date formats	p25	p252
Global variables and parameter passing explicitly included.	Sharing/passing variables between modules.	p25	Making the same data available to different modules: global variables, parameter passing.	p25	p256 p201 (Ch 4)
Functions and procedures specified as different implementations of subroutines.			Use of functions and procedures (Refer p54-56 of Course Specifications)	p25	p256 p202 (Ch 4)
Storyboards added to prelim			Use of storyboard: shows the general design of each interface, shows navigation between interfaces	p25	p262
Navigation and appropriate language added under effective user interfaces.			Navigation Appropriate language for the intended audience	p25	p262 p25 (Ch 1)
Documentation "learn to"s expanded to provide extra guidance	Document code for different audiences, fully document a solution that has been developed in the classroom, use application packages to document a solution, interpret code and documentation prepared by others	p24	Produce documentation for different audiences: produce source code which is well documented and therefore easy to read, understand and maintain. Fully document a solution that has been developed in the classroom. Create a data dictionary to define the data (including variables, arrays and records) used in a developed solution. Use a range of application packages to develop the various types of documentation to fully document a solution. Interpret code and documentation prepared by others. Assess the effectiveness of online help available in software packages	p25 p26	p274 Integrated throughout each chapter.

8.2.3 Testing and evaluating software solutions

Minor changes to emphasise desk checks and comparison of results with expected outputs.

<i>Description of change</i>	<i>Old Syllabus deletions</i>		<i>New syllabus (Preliminary classes from 2011 onwards) additions</i>		<i>Davis SDD Prelim Text Second Edition</i>
Topic name changed to reflect HSC course	Checking software solutions	p25	Testing and evaluating software solutions	p27	p281
Test data outside the expected values now included.			Data which is outside the expected values	p27	p242 (Ch 5) p249 (Ch 5)
Desk checks and comparing results with expected output emphasised.			Compare the actual output from a piece of code with the expected output from test data to detect logic errors Perform a desk check by producing a table showing the changes to the content of variables as the algorithm or code is stepped through manually	p27	p282 p287

8.2.4 Maintaining software solutions

Minor changes to emphasise internal documentation and copyright issues.

<i>Description of change</i>	<i>Old Syllabus deletions</i>		<i>New syllabus (Preliminary classes from 2011 onwards) additions</i>		<i>Davis SDD Prelim Text Second Edition</i>
Topic name changed to reflect HSC course	Modifying software solutions	p27	Maintaining software solutions	p29	p307
Use of white space in source code emphasised.			Appropriate use of white space to improve legibility of the source code	p29	p318
Meaningful identifiers emphasised			Meaningful names for subroutines and modules	p29	p317
Copyright replaces plagiarism.	Plagiarism	p27	Copyright issues Provide adequate acknowledgement of the code of other programmers that has been incorporated as part of the maintenance process	p30	p313
Compatibility with existing code emphasised.			Compatibility of code Assess the compatibility of code to be included in the source code of an existing solution	p30	p313

8.3 Developing software solutions (20%)

Project management has been moved to the start of the topic which better reflects its importance in regard to student projects. This is followed by a list of the system models and other documentation types that are now included in the preliminary course. The developing software solutions subtopic provides a good summary of the significant tasks performed during each phase of the software development cycle.