The basic idea for what to prioritize is to understand principles and not details --- however, what is a detail to some might be a principle to me. Why programs which exhibit locality of references can exploit cache memories better is a principle --- opcodes, the number of rename registers in each register file of the 970MP, and library function parameter types and names are details. Why superscalar processors need branch prediction, rename registers, and a reorder buffer are principles.

181-207	167-196	S	Multicores
211-257	199-240	3	What you can do with the tools but no details such as
			commands or options.
		S	6.8.2 on Helgrind, covered by EDAN26
261-324	243-304	S	Chapters 7 and 8 are covered in EDAA25 C Programming
326-355	305-334	4	Declarations, 9.1 - 9.4.3
356	334	S	9.4.4
356-363	334-341	5	9.5 - 9.6
363-380	341-358	3	9.7 - 9.10
381-382	358-359	S	9.11
382-383	359-360	3	9.12
385-422	361-398	3	10-11 but I will not ask about e.g. precedences
423-440	399-415	S	read only what you need for the projects
441-618	417-580	S	read only what you need for the projects
621-626	581-588	3	14.1 - 14.3
626-648	588-608	4	14.4 - 14.10 but see exceptions below
639-642	600-603	5	14.6.6 - 14.7
649-676	609-637	5	15
677-677	637-637	3	16
678-679	638-639	S	covered by EDAF35
680-685	640-644	5	16.2
685-690	645-649	3	16.3 - 16.4
691-722	651-681	S	covered by EDAN26

2nd ed.	1st ed.	Grade	
1-112	1-103	3	Introductory C
83-86	79-81	4	Matrix memory allocations
125-130	113-118	3	Integer numbers
130-133	118-121	4	Floating point numbers: 2.4 - 2.4.2
133-136	121-123	5	2.4.3-2.4.4
137-164	125-152	3	Power processors but no details specific to only Power!
			You don't need to write assemler code on the exam.
167-179	153-166	3	Cache memories but not the table in Section 4.5.
181-207	167-196	S	Multicores
211-257	199-240	3	What you can do with the tools but no details such as
			commands or options.
		S	6.8.2 on Helgrind, covered by EDAN26
261-324	243-304	S	Chapters 7 and 8 are covered in EDAA25 C Programming
326-355	305-334	4	Declarations, 9.1 - 9.4.3
356	334	S	9.4.4
356-363	334-341	5	9.5 - 9.6
363-380	341-358	3	9.7 - 9.10
381-382	358-359	S	9.11
382-383	359-360	3	9.12

Below are advice on what is important to achieve a certain grade, i.e. if you wish to pass you should study at least those pages marked as 3.

3 = everybody should know this

4 = more advanced

5 = even more advanced

S = skip, e.g. due to it's covered by another course

2nd ed.	1st ed.	Grade	
1-112	1-103	3	Introductory C
83-86	79-81	4	Matrix memory allocations
125-130	113-118	3	Integer numbers
130-133	118-121	4	Floating point numbers: 2.4 - 2.4.2
133-136	121-123	5	2.4.3-2.4.4
137-164	125-152	3	Power processors but no details specific to only Power!
			You don't need to write assemler code on the exam.
167-179	153-166	3	Cache memories but not the table in Section 4.5.
181-207	167-196	S	Multicores
211-	199-240	3	What you can do with the tools

257			but no details such as commands or options.
		S	6.8.2 on Helgrind, covered by EDAN26
261-324	243-304	S	Chapters 7 and 8 are covered in EDAA25 C Programming
326-355	305-334	4	Declarations, 9.1 - 9.4.3
356	334	S	9.4.4
356-363	334-341	5	9.5 - 9.6
363-380	341-358	3	9.7 - 9.10
381-382	358-359	S	9.11
382-383	359-360	3	9.12
385-422	361-398	3	10-11 but I will not ask about e.g. precedences
423-	399-415	S	read only what you need for the

440		projects
441-618	417-580 S	read only what you need for the projects
621-626	581-588 3	14.1 - 14.3
626-648	588-608 4	14.4 - 14.10 but see exceptions below
639-642	600-603 5	14.6.6 - 14.7
649-676	609-637 5	15
677-677	637-637 3	16
678-679	638-639 S	covered by EDAF35
680-685	640-644 5	16.2
685-690	645-649 3	16.3 - 16.4
691-	651-681 S	covered by EDAN26

722