Computer Science - Personal Learning Checklist

Paper 1 Computer Systems

	System Architecture	Difficulty	R	A	G
Computer Systems	1.1 System Architecture	H			
	1.1.1 Architecture of CPU	H			
	1.1.2 CPU Performance	M			
	1.1.3 1.1.3 Embedded Systems	E			
	Memory and Storage	**Difficulty**	R	A	G
	1.2 Memory and Storage	M			
	1.2.1 Primary Storage (Memory)	E			
	1.2.2 Secondary Storage	M			
	1.2.3 Units, Conversion, Capacity and Calculations	H			
	1.2.4 Data Storage	M			
	1.2.4 Data Storage (Numbers)	M			
	1.2.4 Data Storage (Characters)	M			
	1.2.4 Data Storage (Images)	H			
	1.2.4 Data Storage (Sound)	H			
	1.2.5 Compression	M			
	Computer networks, connections and protocols	**Difficulty**	R	A	G
	1.3 Computer networks, connections and protocols	H			
	1.3.1 Networks and topologies	E			
	1.3.2 Wired and wireless networks, protocols and layers	M			
	1.3.2.conn Modes of Connection and Wired and Wireless	M			
	1.3.2.wire Wireless Encryption	E			
	1.3.2.ipmac IP and MAC addressing	M			
	1.3.2.std Standards	H			
	1.3.2.prot Protocols	H			
	1.3.2.lay Concept of Layers	H			
	Network Security	**Difficulty**	R	A	G
	1.4 Network Security	M			
	1.4.1 Threats to computer systems and networks	M			
	1.4.2 Identifying and preventing vulnerabilities	M			
	System Software	**Difficulty**	R	A	G
	1.5 System Software	E			
	1.5.1 Operating Systems	M			
	1.5.2 Utility Software	M			
	Ethical, legal, cultural and environmental impacts of digital technology	**Difficulty**	R	A	G
	1.6 Ethical, legal, cultural and environmental impacts of digital technology	M			
	1.6.1 Impacts of Digital Technology in society	M			
	1.6.1.inv Investigate CS Technolgies	M			
	1.6.1.priv Privacy Issues	M			
	1.6.1.cult Cultural Implications	M			
	1.6.1.env Environmental Implications	M			
	1.6.1.dig Impact of digital technologies	M			
	1.6.1.leg Impact of digital technologies	M			
	1.6.1source Open vs Closed Source	M			

Paper 2 Computational thinking, algorithms and programming (J277/02)

Computational thinking, algorithms and programming (J277/02)

Algorithms	Difficulty	R	A	G
2.1 Algorithms	M			
2.1.1 Computational Thinking (Abstraction, Decomposition)	E			
2.1.2 Designing, creating and refining algorithms	M			
2.1.2.ipo Inputs, Processes and Outputs	E			
2.1.2.stru Structure Diagrams	E			
2.1.2.flow Flow Diags and Pseudo code	M			
2.1.2.err Identifying Errors and Fixing	M			
2.1.2.trace Trace Tables	M			
2.1.3 Searching Algorithms	H			
2.1.4 Sorting algorithms	H			
Programming Fundamentals	**Difficulty**	R	A	G
2.2 Programming Fundamentals	H			
2.2.1 Programming Fundamentals	E			
2.2.1.var Variables, Constants, Input/Output, Assignments	E			
2.2.1.seq Sequence	E			
2.2.1.sel Selection	M			
2.2.1.iter Iteration	H			
2.2.1.ops Arithmetic and Comparison operators	E			
2.2.1boo Boolean Operators	E			
2.2.2 Data Types and Casting	M			
2.2.3 Additional programming techniques	H			
2.2.3.str Strings	M			
2.2.3.arr Arrays	M			
2.2.3.sub Sub Programs	H			
2.2.3.ran Random Number Generation	M			
2.2.3.FIL File Handling	M			
2.2.3.rec Records	M			
2.2.3.sql SQL	M			
Producing Robust Programs	**Difficulty**	R	A	G
2.3 Producing Robust Programs	M			
2.3.1 Defensive Design	M			
2.3.2 Testing	M			
Boolean Logic	**Difficulty**	R	A	G
2.4 Boolean Logic	M			
2.4.1 Boolean Logic	M			
Programming languages and Integrated Development Environments	**Difficulty**	R	A	G
2.5.1 Languages				
2.5.2 Integrated Development Environment				