COMP3520 Operating Systems Internals Assignment 1 Part 1 Marking Scheme

Source Code

Criteria	Mark /10
• Demonstrates mastery of relevant synchronization concepts • Successfully implements an optimal solution that fully meets the assignment requirements with no errors or omissions • Codes with expertise, demonstrating exemplary skills in producing correct human-readable source code that compiles on the School of Computer Science servers	10
• Demonstrates thorough understanding of relevant synchronization concepts • Successfully implements an optimal solution that meets the basic coding requirements with, at most, two minor errors or omissions • Demonstrates excellent skills in producing human-readable source code that compiles on the School of Computer Science servers	9
• Demonstrates broad detailed understanding of relevant synchronization concepts AND • Implements an optimal or near-optimal solution that meets the basic coding requirements but with a few minor errors or omissions AND • Demonstrates excellent skills in producing human-readable source code that compiles on the School of Computer Science servers OR • Demonstrates thorough understanding of relevant synchronization concepts AND • Successfully implements an optimal solution that fully meets the assignment requirements with, at most, two minor errors or omissions AND • Demonstrates well-developed skills in producing human-readable source code that compiles on the School of Computer Science servers	8
• Demonstrates broad detailed understanding of relevant synchronization concepts AND • Implements an optimal or near-optimal solution that meets the basic coding requirements but with a few minor errors or omissions AND • Demonstrates well-developed skills in producing human-readable source code that compiles on the School of Computer Science servers OR • Demonstrates thorough understanding of relevant synchronization	7

concepts AND • Successfully implements an optimal solution that fully meets the assignment requirements with, at most, two minor errors or omissions AND • Produces source code that that compiles on the School of Computer Science servers but that is not always readily human-readable	
• Demonstrates satisfactory understanding of relevant synchronization concepts AND • Implements a sub-optimal solution to the problem, containing some errors or omissions but which meets the basic coding requirements AND • Demonstrates well-developed skills in producing human-readable source code that compiles on the School of Computer Science servers OR • Demonstrates broad detailed understanding of relevant synchronization concepts AND • Implements an optimal or near-optimal solution that meets the basic coding requirements but with a few minor errors or omissions AND • Produces source code that compiles on the School of Computer Science servers but that is not always readily human-readable	6
• Demonstrates satisfactory understanding of relevant synchronization concepts • Implements a sub-optimal solution to the problem, containing some errors or omissions but which meets the basic coding requirements • Produces source code that compiles on the School of Computer Science servers but that is not always readily human-readable	5
• Demonstrates limited understanding of relevant synchronization concepts • Implements a flawed solution to the problem, containing substantial errors or omissions • Demonstrates basic programming skills, producing flawed source code that compiles on the School of Computer Science servers	4
• Demonstrates elementary understanding of some relevant synchronization concepts • Implements a fatally flawed solution to the problem, containing major errors or omissions • Demonstrates genuine engagement with the set programming problem but limited programming skills, producing fatally flawed source code that compiles on the School of Computer Science servers	3
• Demonstrates rudimentary understanding of some relevant synchronization concepts AND • Produces incomplete source code that compiles on the School of Computer Science servers and that contains some evidence of superficial engagement with the set programming problem OR	2

• Demonstrates some understanding of some relevant synchronization concepts AND • Produces relevant source code that does not compile on the School of Computer Science servers but that contains some evidence of genuine engagement with the set programming problem	
• Demonstrates minimal or no understanding of relevant synchronization concepts AND • Produces seriously incomplete source code that compiles on the School of Computer Science servers and that contains minimal evidence of superficial engagement with the set programming problem OR • Demonstrates rudimentary understanding of some relevant synchronization concepts AND • Produces incomplete source code that does not compile on the School of Computer Science servers and that contains some evidence of superficial engagement with the set programming problem	1
• Disqualified by the COMP3520 examiner for any ONE of the following behaviours: o Engaging in an aggravated non-serious attempt in the source code o Engaging in a virtual non-attempt in the source code o Failing to submit source code OR • Disqualified by the Faculty of Engineering or the University due to academic dishonesty or misconduct in this assignment	0