COMP3301/COMP7308 Assignment 2 Marksheet 25 Marks (25%)				
$Revision: 486 $				
Surname	**First Name**	**Student ID**	**Session/Date**	**Marker Initials**

Criteria	Excellent (eX)	Good (gX)	Satisfactory (sX)	Weak (wX)	Poor (pX)	
Functionality	• (e1) Implements an `autoconf(9)` device driver that can match and attach to the A2 device. • (e2) Implements a character device driver at major specified. • (e3) Correctly implements `open()`, `close()`, and `ioctl()` interfaces as specified. • (e4) Correctly sets up and utilises the device descriptor rings to send and receive commands and responses, including under concurrent load.	• (g1) Implements an `autoconf(9)` device driver that can match and attach to the A2 device. • (g2) Implements a character device driver at major specified. • (g3) Implements `open()`, `close()`, and `ioctl()` interfaces as specified, with minor deviations in error cases or mistakes that do not impede basic operation. • (g4) Sets up and utilises the device descriptor rings to at least complete command and response cycles submitted serially, but may fail under concurrency.	• (s1) Implements an `autoconf(9)` device driver that can match and attach to the A2 device. • (s2) Implements a character device driver at major specified. • (s3) Implements `open()`, `close()` interfaces as specified. Implements enough of `ioctl()` to exchange at least one command with the device. • (s4) Sets up and utilises the device descriptor rings to successfully submit and complete at least one command.	• (w1) Implements an `autoconf(9)` device driver that can match and attach to the A2 device. • (w2) Implements a character device driver at major specified. • (w3) Implements at least `open()` and `close()`, and these work at least once. Some attempt to implement `ioctl()`. • (w4) Maps the device BAR and attempts to allocate and configure descriptor rings, but may not be successful.	• (p1) Attempts to implement an `autoconf(9)` device driver that can match and attach to the A2 device. • (p2) May or may not implement a character device driver at specified major. • (p3) May or may not implement `open()`, `close()` and `ioctl()` for the character device, if attempted. • (p4) Attempts to at least map and use the device BAR.	• (z1) No code submitted, or code does not compile and boot. • (z2) No attempt to implement an `autoconf(9)` device driver (no match/attach). • (z3) No evidence of an attempt to map or use device BAR or descriptor rings.
/12	12	11 – 9	8 – 6	5 – 3	2 – 1	0

Criteria	Excellent (eX)	Good (gX)	Satisfactory (sX)	Weak (wX)	Poor (pX)	
Robustness	• (e1) Hardware errors are detected in an efficient and timely manner, according to spec, and returned to userland where necessary. • (e2) Errors in arguments or data from userland are handled appropriately and consistently, according to spec. • (e3) Driver code checks for errors on all memory allocations and calls to other parts of the kernel. • (e4) Driver thoroughly qualifies which devices it attaches to and will not attach erroneously to a non-A2 device. • (e5) Lengths of memory allocations and copy operations are calculated safely and checked. • (e6) Functions are consistently designed to release resources correctly and return on error. • (e7) Extensive use of defensive programming is evident in checks for expected structure and values and use of assertions. • (e8) Locks and related concurrency primitives are used appropriately and consistently.	• (g1) Hardware errors are detected appropriately, perhaps with a few small deviations from spec. • (g2) Errors in arguments or data from userland are handled appropriately, perhaps with a few small deviations from spec. • (g3) Driver code checks for errors on memory allocations and calls to other code, perhaps with a few isolated exceptions. • (g4) Driver qualifies which devices it attaches to and will not attach erroneously to a non-A2 device. • (g5) Lengths of memory allocations and copy operations are calculated safely and checked. • (g6) Functions are consistently designed to attempt to release resources and return on error. • (g7) Some use of defensive programming is evident. • (g8) Locks and related concurrency primitives are used appropriately.	• (s1) An attempt is made to detect hardware errors in a sensible way, but it may be flawed. • (s2) Consistent attempts are made to handle errors in arguments or data from userland, and mostly to spec. • (s3) Driver attempts to qualify which devices it attaches to and is mostly successful. • (s4) Driver code mostly checks for errors on memory allocations and external calls. • (s5) Lengths of memory allocations and copy operations are calculated safely and checked • (s6) Resources may be leaked in error cases, but success cases are mostly handled appropriately/ • (s7) Attempted to use at least one defensive programming technique. • (s8) Attempted to use locks in a sensible way.		• (p1) Hardware errors handled inconsistently or inappropriately. • (p2) Errors in arguments or data from userland are not detected or handled inappropriately. • (p3) Driver attempts to qualify which devices it attaches to. • (p4) Missed checks on errors from memory allocation or external calls. • (p5) Errors in length calculations which could result in overflow or unsafe operation. • (p6) Some attempt at functions being design to release resources and return on errors. • (p7) Little or no evidence of defensive programming. • (p8) Little or no evidence of concurrency concerns or attempt to deal with them.	• (z1) No code submitted, or code does not compile and boot.
/8	8	7 – 5	4 – 2	–	1	0

Criteria	Excellent (eX)	Good (gX)	Satisfactory (sX)	Weak (wX)	Poor (pX)	
Style	• (e1) No style errors were detected in submitted code.	• (g1) A small number of isolated (one-off) style errors were detected.	• (s1) Widespread style errors, including any repeated, consistent violations of the same rule.		• (p1) Code was submitted, and some lines of code contained no style violations.	• (z1) No code submitted, or every line submitted contains style violations.
/5	5	4 – 3	2	–	1	0

Total	_____ / 25
Comments	