Rubrics & Grading criteria

To pass a level for a specific rubric you also need to pass all the 'lower' levels in that rubric *and be able to explain all your code*.
All clients for assignment 2, 3 & 4 must be implemented in Unity, all servers are either console based or implemented in Unity (console based is advised).

Criterium / learning goal	**Sufficient (5.5)**	**Good (7)**	**Very good (8.5)**	**Excellent (10)**
Learning goal 1 (Assignment 1) Student recalls basic TCP/IP theory and principles.	All 'sufficient' questions answered correctly.	All 'good' questions answered correctly and handed in on blackboard **before** week 3.	All 'very good' questions answered correctly and handed in on blackboard **before** week 2.	All 'excellent' questions answered correctly and handed in on blackboard **before** week 2. Perfect, you clearly know your stuff!
Learning goal 2 (Assignment 2, 3 & 4) Student uses reliable (object-based) messaging to create a networked application.	Student meets all the 'sufficient' requirements for assignment x • Code uses clear variable names and structure. • Server never crashes while clients play nice (but they might leave).	+Student meets all the 'good' requirements for assignment x • Code is readable and split correctly into readable/ reusable methods.	+Student meets all the 'very good' requirements for assignment x • Student clearly invested time in code setup and architecture • Applications provide extensive debugging info (e.g. protocol messages can be printed).	+Student meets all the 'excellent' requirements for assignment x • (Almost) bug free.
Assignment / LG / %grade / ±hrs				
Assignment 1 (1) 10% (6)				
Assignment 2 (2.1-2.3) 20% (12)				
Assignment 3 (2.1-2.4) 30% (18)				
Assignment 4 (2.1-2.5) 40% (24)				