

1A Logic: Worksheet 4	5	<i>Excellent</i>	
	4	<i>Good</i>	
Your name:	3	<i>Satisfactory</i>	
Logic class (A/B/C/D/E):	2	<i>Weak</i>	
Logic class tutor:	1	<i>Very poor</i>	

Reading

Read *Introduction to Formal Logic*, Chapters 16–18.

Do take careful note of §17.4 which explains a point (about how to test for tautologies) that was only rather rapidly made, without pausing over any examples, in lectures.

Then do the following exercises as instructed, and firmly clip/staple this question sheet – with grid correctly completed – onto the front of your work (include your work for the self-marked Section 1).

1 Exercises from the Book

Do the following questions from the end-of-chapter exercises in *An Introduction to Formal Logic*. Then, when you have completed them, carefully check your answers against the answers available at www.logicmatters.net. Correct your own work *in red*, for the marker to review. In the box below, note any residual queries or problems you have with these self-marked exercises (use a continuation sheet if you have more queries than you can mention here). Take disjunctions to be inclusive!

Exercises 17 (p. 170): Qns A2, A3, A5, A7, A8, A10

Exercises 17 (p. 170): Qns B2, B4

Exercises 18 (p. 178): Qns A1, A2, A8, A9, A12.

Queries

Is there a continuation sheet with more queries? Yes/No

2 Further exercises

Practice makes perfect! (Let's say it again: there's nothing very exciting about PL trees – the point of practising them hard is that once you can do PL trees without too much thought, you should very easily be able to extend your technique to cope with QL trees.)

1. Use trees to determine which of the following are true (for the significance of the sign ' \vDash ' see again §13.7 and §14.6).

(a) $((P \vee Q) \supset R) \vDash ((P \supset R) \vee (Q \supset R))$

(b) $\vDash \neg(((P \supset Q) \wedge (Q \supset P)) \wedge \neg(P \equiv Q))$

(c) $(P \equiv (\neg R \vee \neg Q)), \neg(\neg Q \vee \neg S), (R \equiv S) \vDash (P \wedge S)$

2. Explain carefully what the relation is between an argument's being (a) valid, (b) valid by the truth-table test, (c) valid by the tree test.