COMS10014 Class Test 1 Solutions

Question 1

The general idea is that a term can contain only constants, variables, logical operations and brackets, and must parse correctly. Something that would be a term except that it contains exactly one statement symbol ($\equiv, \vDash, \vdash$ etc.) as the top-level operation (where a binary operation would be allowed), is a statement. Anything else, for example more than one statement symbol, is invalid.

1.	$x \vDash x \lor y$	is a statement, but not a term, as it contains a $\vDash$ and it parses correctly.
2.	$(x \lor y) \to (x \land y)$	is a term, as it only contains logical operations, and parses correctly (it is not a tautology, but that was not asked for here).
3.	$(x \equiv y) \to x$	is invalid, as statement symbols such as $\equiv$ can only appear as the top-level operation, not as part of a term.
4.	$(x \neg y) \land (x \lor y)$	is invalid, as $\neg$ is a unary not a binary operation, so this string does not parse.
5.	$x \lor y \equiv y \lor x$	is a statement. The $\equiv$ is the top-level operation.
6.	$(x \land y) \vdash x \vdash (x \lor y)$	is invalid, as there is more than one statement symbol.

Question 2

Both $p \uparrow p$ and $p \downarrow p$ are equivalent to $\neg p$. So,

1. is $\neg(\neg p \land \neg q)$ which, by DeMorgan and double negation, is $p \lor q$ so A.
2. is $\neg(\neg p \lor \neg q)$ which, by DeMorgan and double negation, is $p \land q$ so B.

The other two are of the form $X \downarrow X$ resp. $X \uparrow X$ for $X = p \downarrow q$, but we know from above that both these expressions must be equivalent to $\neg X$. By double negation, $\neg\neg(p \lor q)$ is $p \lor q$ so 3. and 4. are both A. again.

Question 3

1. False. This is the converse, not the contrapositive.
2. False. An implication is equivalent to its contrapositive, not its converse.
3. True. The negation of the converse can only be true if the antecedent (of the original implication) is false, in which case the implication is true.
4. True. The inverse of $A \to B$ is $\neg A \to \neg B$
   (we are eliminating the double negation in "is not not a *Boojum*" that you get this way).
5. False. An implication is equivalent to its contrapositive – not the negation thereof.
6. False. For the original $A \to B$, the converse would be $B \to A$, but here we have $A \to \neg B$.

Question 4

1. This is true. $P \vDash Q$ means that every column of a truth table in which P is true, also has Q true. That means there is no column where $P \to Q$ could be false, as that only happens for $P = T, Q = F$, so $P \to Q$ is a tautology. Conversely, if $P \to Q$ is a tautology, then this means the case $P = T, Q = F$ cannot happen (for example because P, Q share some underlying variables), which means that in every column where P is true, Q is true too.
2. This is true, because the laws of logic (as presented) are complete and sound.
3. This is true, as one can check with a truth table. $P \leftrightarrow Q$ is true if and only if P, Q both have the same value, and in both cases $P \to Q$ is true also.
4. This is false. For $P = T, Q = F$ we have $P \to Q \equiv F$ but $P \lor \neg Q \equiv T$.
   The formula $P \to Q \equiv \neg P \lor Q$ would be true, but that is not what was asked here.
5. This is false. A contingency has at least one assignment that makes it false, so its negation is true for this assignment. In fact, the negation of a contingency is always a contingency again, but negating swaps tautologies and contradictions.
6. This is true. Pick any row where P is true in the truth table (over all variables involved in the expressions). The by $P \vDash Q$, in this row Q is also true, and by $Q \vDash R$, also R is true in this row. But that is the definition of $P \vDash R$.

Question 5

Columns 1, 2, 5, 6 can just be copied from the left (you don't even need to write these out if you don't want to). Column 3 is the negation of column 2; column 4 is the and of columns 1 and 3, column 7 is the implication of columns 5 and 6, and column 8 is the conjunction of columns 4 and 7. Column 8 is the final answer.

The method is to cross off columns as they get taken off the stack, then whenever you see an operator, it applies to the last two (last one if unary) columns that are not crossed off yet. So, when we encounter column 3, we cross off column 2 since negation in RPN is pop one item, negate it, and push the result back on the stack. Column 4 then has two columns left on the stack, 1 and 3, so we take the conjunction of these and cross them both off. Column 7 crosses off columns and 6, so when we come to column 8, only columns 4 and 7 are still uncrossed, so these are the ones we need for the last column.

			1	2	3	4	5	6	7	8
p	q	r	p	r	$\neg$	$\land$	q	p	$\to$	$\land$
F	F	F	F	F	T	F	F	F	T	**F**
F	F	T	F	T	F	F	F	F	T	**F**
F	T	F	F	F	T	F	T	F	F	**F**
F	T	T	F	T	F	F	T	F	F	**F**
T	F	F	T	F	T	T	F	T	T	**T**
T	F	T	T	T	F	F	F	T	T	**F**
T	T	F	T	F	T	T	T	T	T	**T**
T	T	T	T	T	F	F	T	T	T	**F**