Quiz Section 1: Propositional Logic Translation

Task 1 – Stop, Prop, and Roll [18 pts]

Translate these English statements into logic, making the atomic propositions as simple as possible and exposing as much of the logic via symbols as possible.

a) Define a set of *at most four* atomic propositions. Then, use them to translate "If I haven't had my coffee and the sun is not up, then I am angry. But if I have had my coffee or the sun is up, then I am happy."

b) Define a set of *at most four* atomic propositions. Then, use those propositions to translate each of these sentences:

i) If the stack is empty, you can push but not pop.

ii) If the stack is full, you can pop but not push.

iii) If the stack is neither full nor empty, you can both pop and push.

c) Define a set of *at most four* atomic propositions. Then, use them to translate each of these sentences:

i) You can have your cake or you can eat your cake, but not both.

ii) If you have your cake and you drop your cake, then you are sad and you don't have your cake. But if you eat your cake and are sad, then you don't have your cake.

Task 2 – The Calm Before the Form [14 pts]

The Java project you are working on contains the following function. It takes four boolean arguments, a, b, c, and d, and calculates some boolean function of them called E:

```
public static boolean E(boolean a, boolean b, boolean c, boolean d) {
  if (!(a || b))
    return false;
  if (!(!a || !b))
    return false;
  if (!(a || c))
    return false;
  return (b || !d);
}
```

The code checks the value of the four disjunctions $a \lor b$, $\neg a \lor \neg b$, $a \lor c$, and $b \lor \neg d$ and returns true iff *all four* of them are true. In other words, it calculates the conjunction (and) of those four expressions. Specifically, it calculates the same value as the following (non-canonical) CNF expression:

$$(a \lor b) \land (\neg a \lor \neg b) \land (a \lor c) \land (b \lor \neg d)$$

Note that this Java code could be written equivalently as a single `return` statement:

```
return (a || b) && (!a || !b) && (a || c) && (b || !d);
```

The Java compiler would automatically translate that into the series of `if` statements above that returns `false` as soon as one disjunction is found to evaluate as false, a process known as "short circuiting".

a) Write a truth table for E. Include columns for a, b, c, d, all four disjunctions, and E.

b) Write the **canonical** DNF expression for E.

c) Translate your DNF expression into a new Java implementation of `E`.

Like above, your code should be a series of `if` statements and then a `return`, except that, now, each `if` should check the value of a conjunction and return `true` if it is true.

(Do not simplify. Translate your expression to code in the most direct way possible.)

Task 3 – Welcome to the $\forall\mathrm{LL}\exists n$ School! [16 pts]

Let the domain of discourse be all students and classes at UW. Define the predicates CS(x) to mean that x majors in CS (presumably x is a student) and CE(x) to mean that x majors in CE (presumably x is a student). Define the predicates CSE(y) to mean that y is a CSE class (presumably y is a class) and MATH(y) to mean that y is a MATH class (presumably y is a class). Define the predicate Wants(x, y) to mean that x wants to take y (presumably x is a student and y is a class), the predicate Likes(x, y) to mean that x likes y (presumably x is a student and y is a class), and the predicate HasToTake(x, y) to mean that x has to take y (presumably x is a student and y is a class).

Translate each of the following logical statements into English. You should not simplify. However, you should use the techniques shown in lecture for producing more natural translations when restricting domains and for avoiding the introduction of variable names when not necessary.

a) $\neg\exists x\,(\mathsf{CS}(x) \land \mathsf{CE}(x))$

b) $\exists x\,\big(\mathsf{CS}(x) \land \exists y\,(\mathsf{CSE}(y) \land \neg\mathsf{HasToTake}(x, y) \land \mathsf{Likes}(x, y))\big)$

c) $\forall x\,\big(\mathsf{CE}(x) \to \exists y\,(\mathsf{MATH}(y) \land \mathsf{HasToTake}(x, y))\big)$

d) $\exists x\,\big((\mathsf{CS}(x) \lor \mathsf{CE}(x)) \land \forall y\,(\mathsf{CSE}(y) \to \mathsf{Wants}(x, y))\big)$