What can a philosopher and logician do for your enterprise?
v. 3,

A series of un-ordered top ten lists with brief, non-exhaustive annotations to answer the question above follows.

Top ten skills concerning philosophy in general:

1. Strong vocabulary - high score on GRE, exposure to the vocabulary of many fields.

2. Familiarity with many sorts of documents - monographs, meditative essays, dialogues, technical reports, etc. Philosophy includes these and others.

3. Historical sensitivity - Philosophers often spend almost too much time looking at the past of their field. Aristotle and Hume are often treated as contemporaries. Learning how something came to be helps to understand its current (dis)function.

4. Analytical mindset - Philosophers are trained to make distinctions between apparently similar items. For example of this in a whimsical vein, see Frankfurt, *On Bullshit*.

5. Writing and revision - Philosophers write, for each other, for the public and for students. These differing audiences yield a familiarity with many writing styles, revision techniques, etc.

6. “Big picture” view - connections between apparently disparate items (i.e., synthesis). Historical, conceptual and other links are often remembered, found or created by philosophers. The philosopher is an intellectual generalist, and a “flyer between” various other disciplines.

7. Ethical systems and principles - philosophers are trained in ethics. While this might not make us more moral than anyone else, it at least allows us to articulate different ways of understanding ethical systems, to develop ways to encourage articulation of ethical values, and to develop a sensitivity as to where values “get located”.

8. Argumentation and debate - the notion of “dialectic”, going back to Socrates, is also vital. We know how to defend our opinions, see the weaknesses in others, and yet also find common ground for fruitful discussion.

9. Sensitivity to language and word choice - being analytical (item 4) permits one to see subtle difference in word usage in ordinary language, and, indeed, sometimes also in technical contexts. I have investigated (informally at this point) usage of the notion of “stability” in chemistry, for example. We are also keenly aware of the difference between logic and rhetoric, and know about the “conversational implicatures” of ordinary language that some people either miss or are unaware they are making.

10. Tools to interrelate cultures and views from around the world - Logic is universal in the sense that it can be used by anyone and in any subject whatsoever. The “space” of metaphysical positions is quite well known and can be used to help articulate the views of those who might wish to communicate their own.

Top ten skills concerning logic in specific:

1. Mathematical familiarity (proofs, etc.) - While we are not mathematicians, someone who has done some graduate level courses in logic has seen mathematics as understood by mathematicians both as a tool and as an object of metastudy and reflection.

2. Rigorous reasoning - Formalism allows one to be extra careful when reasoning, by making sure some of the processes are absolutely mechanical, or at least, rigorously justified. I routinely see errors in logic in the news. For example, the recent (at the time of writing) flap about a joke by John Kerry. What he said, according to his critics, is formalized “if you don’t study hard, you’ll end up stuck in Iraq”. His critics took him as saying that the soldiers in Iraq didn’t study hard, etc. This is an example of affirming the consequent, an elementary logical error. Even according to his critics, Kerry did not say “if and only if ...”

3. Abstraction - Logic, like any tool of analysis (mathematical or otherwise), encourages (and requires) discarding what is perceived to be inessential to focus on what is important. This process is explicit in logic and so reminds us of the strengths of the process (clarity of thought, etc.), and the potential pitfalls (being mistaken about what is to be ignored, etc.)

4. Thinking at several levels of analysis - In a second or third course in logic, one often starts thinking about metatheoretic concerns. These are properties of a given logical system, such as completeness, etc. This area of inquiry requires one to think about “the inside” and “the outside” of a system and keep them straight in one’s mind at all times on pain of horrible confusion. This training can be applied to other considerations where it is useful, for example in ethics, policy choice, etc.

5. Symbolization - Being trained to represent an argument, proposition or other item currently in natural language in symbols is useful both for abstraction (see point 3), but also as a useful shorthand in other contexts. This makes one unintimidated by symbols and allows one to more easily see through bafflegab or pseudosymbolization.

6. Diagnosis of argument oversights - Most people think of logic in the context of critical thinking, close reasoning and so on. This view is correct, just limited. Nevertheless, this aspect is absolutely vital as far as it goes. (See also point 2.)

7. Understanding of the limitations of computers - Logic techniques reveal to us what our conventional computing devices can and cannot do. For example, it is a

computationally unsolvable problem to discover “dead code” (code unreachable by any possible execution of the program.).

8. Teaching logic - hence teaching and other educational roles generally.
9. Use of boolean algebras and other very general formal tools - Some of the formal aspects of logic have many applications. Direct logical tools are used with profit in philosophy proper, computer architecture, the theory of databases, artificial intelligence, linguistics, cognitive psychology and others. The logician’s familiarity with the “common ground” both makes promotes a teaching merit when it comes to the subject matter of logic itself, as well as allowing him to more easily pick up new uses of the tools.
10. Blending of creativity and mechanical reasoning - Any proof, even a formal one, requires creativity to create a “perspicuous” or at least manageable proof. This may be contrary to how a lot of people think about logical or mathematical reasoning, but it is nevertheless true. (Career testing often opposes creativity to mathemathical or scientific reasoning.)

Top ten skills resulting from my speciality in the philosophy of computing:

1. Familiarity with many computer applications and the ability to learn others quickly - I may not be a computing professional per se, but I have had a large exposure to computing technology and am fearless when it comes to exploring new software (and, in the old days, hardware).
2. Computer programming, science, and technology - I have some knowledge of several classes of languages (procedural, object-oriented, functional, query), data structures and algorithms, file systems, database design, theory of computing, computer architecture, networks, etc. (See the “IT and Me” document.)
3. Humanistic approach to technology - I have studied (both in school and independently) the ethics and aesethics of computing, as well as its political, historical and social sides. This yields a technological approach beyond “let’s build what works and what won’t cost us too much now or later”.
4. Computing as an educational tool - Computing is increasingly being used in the humanities classroom. I try to keep up these trends, e.g. by following the Computing and Philosophy conferences.
5. Historical sensitivity to the development of computing - Some of my reading into how the computer arose that emphasizes the mathematical background rather than the usual discussion of machines. I (re)discovered the falsity of three myths often repeated in computer science circles. Here are the facts: the “Church-Turing” thesis is really something like the “Post-Kleene” thesis, though ironically Kleene probably did much to spread the misnomer himself. (Modesty?) Second, that the “Turing machine” was NOT motivated by reasoning about some hypothetical calculational

device. It was, instead, a result of reflection about an idealized human clerk. In fact, philosophically speaking, Turing made the ingenious step of moving from pure mathematics to metaphysics (or the philosophy of mind and psychology more generally) to provide a solution to the Entscheidungsproblem. Thirdly, as far as I am concerned, one cannot hope to provide a pure mathematical answer to said problem, as it **isn't about math**, but rather about the ability to do math, which is something else.

6. Aesthetic and ethical analysis of computing systems - Not only have I studied specific case studies (see 3) but I am aware of how to conduct specific evaluations myself, as well as have familiarity with general approaches to value-theoretic analysis. For example, Bunge's "rule based on law" principle is one useful approach to get people to "put their values on their sleeves".

7. Analysis of crucial arguments in certain foundational areas of computing - Debates over AI, program verification, the nature of computation, the "computational theories of mind", etc. all have philosophical roots and connections. For example, the current (pseudo?) debate over "hypercomputation" is most profitably seen (in my view) as a debate about the limitations and roles of abstraction and modelling.

8. Framework for understanding computing systems - The theory of computation, value theory systems (see elsewhere in this document), general logical principles, etc. form a good general set of tools to understanding many different aspects of computing

9. Vocabulary for "translating" between computing professionals and others - Being one who is required to maintain "a third way" between two ways of thinking (some may regard as disparate) allows me to develop patience and techniques somewhat profitable in attempting to "translate" between the technical and the humanistic. For example, recently I paraphrased a software developer's remark that he found a C function with 6000 lines of code using the analogy of discovering a chapter in a novel which was all one sentence.

10. Computational approach to problem solving - Daniel Dennett, Clark Glymour and other relatively famous philosophers have long emphasized that the computer makes a wonderful tool for "thought experiments". Often trying to think through (or even doing!) a computer program, spreadsheet, etc. to solve a given problem is a very useful exercise. For example, I wanted to quickly figure out some labourious calculation based on some science fiction story I had heard of; I briefly wrote a short C program to do the calculation for me.