The Abstract Assignment
An abstract describes the basic content of the project to the reader. It includes
1. What you are doing
2. Why you are doing it
3. How you are doing it
4. The relevance of the results and the conclusion.

An abstract is a self-contained single paragraph that describes the work. It should not include abbreviations, acronyms, or bibliographic information. The point is to enable someone unfamiliar with the topic to quickly understand what is being done, and the wider relevance of the work. **Your abstract should be between 400-450 words and must include 3 academic and 2 non-academic sources**.

What to write in an Abstract?

Motivation: A little history about who's done what and how your work fits in with it.
Why do we care about the problem and the results? If the problem isn't obviously "interesting" it might be better to put motivation first; but if your work is incremental progress on a problem that is widely recognized as important, then it is probably better to put the problem statement first to indicate which piece of the larger problem you are breaking off to work on. This section should include the importance of your work, the difficulty of the area, and the impact it might have if successful.

Problem statement: What you're trying to tell the audience that they don't already know (e.g. Your story.)
What problem are you trying to solve? What is the scope of your work (a generalized approach, or for a specific situation)? Be careful not to use too much jargon. In some cases it is appropriate to put the problem statement before the motivation, but usually this only works if most readers already understand why the problem is important.

Approach: Why the audience should believe that the results you've got aren't made up or flawed
How did you go about solving or making progress on the problem? Did you use simulation, analytic models, prototype construction, or analysis of field data for an actual product? What was the extent of your work (did you look at one application program or a hundred programs in twenty different programming languages?) What important variables did you control, ignore, or measure?

Results: Evidence that you've come up with that confirms your story
What's the answer? Specifically, most good computer architecture papers conclude that something is so many percent faster, cheaper, smaller, or otherwise better than something else. Put the result there, in numbers. Avoid vague, hand-waving results such as "very", "small", or "significant." If you must be vague, you are only given license to do so when you can talk about orders-of-magnitude improvement. There is a tension here in that you should not provide numbers that can be easily misinterpreted, but on the other hand you don't have room for all the caveats.

Conclusions: Recap of your story and its implications
What are the implications of your answer? Is it going to change the world (unlikely), be a significant "win", be a nice hack, or simply serve as a road sign indicating that this path is a waste of time (all of the previous results are useful). Are your results general, potentially generalizable, or specific to a particular case?

This text has been mildly modified from Koopman (1997) "How to Write an abstract"
http://www.ece.cmu.edu/~koopman/essays/abstract.html