**Preparing for the ATTA Exam – Next Steps**

In class we’ve covered some essential C# language features, talked about commonly used data structures, looked at selected portions of the .NET Framework libraries, and discussed a methodology for analyzing and solving problems. Here are some general tips for continued preparation, some suggestions for review priorities, and some additional resources you can check out.

**General Practice Suggestions**

1. Try to spend at least a half hour each day writing code. This might not be possible all the time, but like any skill-building you’ll get your best results if you practice consistently. It’s much better to put in an hour or two each day than to cram for ten hours the day before the test.
2. Spend some time reviewing the MSDN documentation on .NET classes that you use regularly. A deep understanding of the tools you use when writing code will serve you well during tests, interviews, and on the job.
3. Apply the methodology we learned in class to every problem you try to solve. Don’t skimp on planning if you’re tackling a simple problem; you might overlook an edge case. If you get stuck on a challenging problem try breaking the problem down, looking at concrete use cases, and simplifying test cases as much as possible. Almost every challenging problem in software engineering is solved by breaking it down into simpler problems.
4. For really challenging situations, don’t be afraid to reach out for help. Send your questions to me, to your managers and mentors, or post your question on a forum like Stack Overflow. Someone somewhere has encountered the same problem as you and knows how to solve it!
5. Read other people’s code. Even if you’re not entirely sure what you’re seeing at first, getting a feel for how more experienced programmers solve problems will clue you in to language features and techniques that you may not presently know about. Research the things you don’t understand and add them to your toolkit!
6. Don’t forget that the debugger is your friend! Use it to understand how your program is working or how it’s failing to work. Be patient and methodical, and take notes if necessary. With a little careful observation you’ll find the error.
7. To reiterate the first point, because it’s the most important: You learn to write code by writing code. Practice daily. Don’t put it off: Your skills will strengthen much more quickly if you stay consistent!

**Review Priorities for the ATTA Examination**

C# is a very robust language and we’ve discussed a lot of concepts in class. Here are some tips for prioritizing your practice:

1. Problem solving is key; look at the review exercises form our past classes and tackle assignments that you haven’t finished or couldn’t work out the first time. Use the resources included below to find new problems!
2. If there are basic language features that you feel unsure about (loops, basic types, flow control, etc.), immediately review these concepts and practice them. Without a solid grasp of the fundamentals everything else is unnecessarily difficult.
3. Revisit common data structures and practice using them. In particular, make sure you’re comfortable with arrays, Lists, Stacks, Queues, and Dictionaries. You are very likely to encounter problems on your ATTA exam that will be much, much easier to solve if you have a solid grasp of when and how to use these crucial classes! Review the MSDN documentation for each of these structures and practice using them.
4. Similarly, make certain that you’re comfortable with common types. Understand what they are and how you should use them. In particular, I recommend reviewing **strings, integers, floats and doubles, chars, and Boolean values.** Read the MSDN documentation on these types if you haven’t already and make sure you have a thorough understanding of them!
5. I don’t think it’s likely that your test will require you to create your own classes or structs, so these concepts shouldn’t be your highest priorities; nevertheless, don’t let them fall completely off your radar. These concepts are crucial to any real understanding of object-oriented programming and will likely come up in an interview.
6. Finally, allot some time for less-crucial concepts or concepts that weren’t covered in class. Here’s a short list of concepts that are worth spending some time with if you haven’t done so already: Enumeration (or enum), regular expressions, mutability, the keywords *out* and *ref*, and delegates. These ideas aren’t likely to be crucial to your success on the ATTA, but they will likely be important to you as you continue to develop your skills.

**Additional Resources**

As always, expect to spend some time reviewing the MSDN documentation. Here’s a good starting point:

<https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/>

If you haven’t already done so, I strongly encourage you to register for a HackerRank account and spend some time practicing there. It’s an awesome way to work on problems of all types in many different programming languages. Definitely make use of this resource!

<https://www.hackerrank.com>

Here’s a little website I like that provides a lot of practice ideas, from very simple to fairly complex. Check it out if you’re looking for additional practice:

<https://www.w3resource.com/csharp-exercises/>

This website is an excellent resource for programmers of all skill levels and interests. You may have to do a bit of searching to find material relevant to your interests, but it’s worth checking out:

<https://www.codeproject.com/>

And finally, I’m going to try to set up some office hours over the next couple of weeks so that those who wish to ask questions or review code can do so. In the meantime, be sure to reach out to me with any questions you have at [semerson@blizzard.com](mailto:semerson@blizzard.com) .