

## Assignment 7

# C# Language Report

**Paper Due:** 11:59 PM, Thursday, December 9, 2004

This assignment gives you experience in applying the ideas you've learned in CS 131 to a new language.

## Instructions

For this assignment you are requested (but not required) to work with another student in the class. If and when you select a partner, edit the Ass7Partners page on the Wiki to indicate who you are working with. If you choose not to work with a partner, move yourself to the "working alone" section of that page. If you need a partner, the Wiki page should provide you with information about who is available.

If you do work with another student, only one person should submit. Although different parts of the report may be originally written by different people, you are both responsible for reading through and editing the entire document, and for making sure that it feels like a complete and coherent whole.

You should `cs131submit` your answer as a PDF file `csharp.pdf` (or, if absolutely necessary, a PostScript file named `csharp.ps`.)

## Scenario

You are consulting for a small software company that is considering switching to the C# programming language. Their programmers are currently spending all of their time getting a release out the door, and thus don't have much time to think about new programming languages. You have been hired to look into C#; the hope is that by reading your report their programmers (who already know C++ and Java) can review this new language more quickly.

Since your contract is short-term, the company has requested that you concentrate on the following specific issues:

- C#, like Java, has a type system that is supposed to ensure safety. What sorts of types does C# provide? [You do not need to list every single type available to the programmer; types that are very closely related (e.g., all object types) can be treated as a group.]
- How are these types related by subtyping (implicit conversions)?
- What kinds of parameter passing does C# support?
- What kinds of genericity does C# support?

- What are the rules for overloading? Specifically, under what conditions can two methods with the same name coexist in the same class? Given specific arguments, how does the system choose which of the available methods is the right one (or decide that none is right)?
- In C++, methods cannot be overridden in subclasses unless they were explicitly declared **virtual** when defined. In contrast, all methods in Java can be overridden in subclasses unless explicitly declared **final**. How do inheritance and overriding work in C#?
- C# includes other constructs called *properties*, *delegates*, and *events*. What are they, what are they good for, and how do they compare to constructs in other languages? [Delegates, in particular, seem to act a little like function values, but are they more like code pointers, closures, or something else entirely?]
- What one other aspect of C# is (in your opinion) most important for the company's programmers to be aware of?

Fortunately, their programmers have all taken a college course in programming languages, and so you can and should use ideas and vocabulary discussed in CS 131.

## Requirements

Your report should:

- Address all the requested issues
- Have clear, accurate, and complete answers.
- Use appropriate vocabulary
- Provide a bibliography of resources consulted
- Provide example code and/or comparisons with other languages, if doing so would improve clarity.
- Be relatively self-contained (e.g., the report should not assume the reader saw the problem statement or has previous knowledge of C#)
- Have no spelling or grammatical errors
- Be legible (i.e., meet the same standards required for your other written work, and thus have sensible line lengths, margins, fonts, and so forth—ideally, it should be written with  $\text{\LaTeX}$ )
- Be your own work (i.e., your report should not contain paragraphs or example code copied verbatim from other sources)

## Resources

You may use any reliable general resources on C#, as long as they are documented in your bibliography. For example, there are a couple (not very good and not very recent) C# books available in the Terminal Room (which must *stay* in the terminal room), Sprague Library has several other books, and there is plenty of very detailed information available online (e.g., from Microsoft directly at <http://msdn.microsoft.com/>). Other resources, including the help system of Microsoft's *Visual Studio* may also be helpful.