Choir Maker

Senior Integration Project

Reagan Duggins

13 April 2016

­

**Introduction: Christian Faith, Software Development… And Music? Tie in your project more**

In Christian education, it is not uncommon for people to consider disciplines like Computer Science to be less Christian, or more “worldly” than disciplines like Theology or Biblical Studies. Though this may seem obvious, is it really the case? Does the act of learning to program a computer inherently bring us farther from, or prevent us from getting as close to, a meaningful relationship with our Creator? Or can we still effectively serve our Lord, even in studies like Computer Science? This paper will endeavor to answer these questions in the light of a Christocentric worldview, and our identities as bearers of the image of Christ.

Computers are powerful tools, there is no question about that. The ability to process billions of pieces of information in mere seconds is amazing, and allows us to do and make amazing things. But as the old saying goes, with great power comes great responsibility. As with any powerful tool, computers can be used both to help and to hinder, to heal and to harm. Computers are neither completely bad nor completely good, though many people make the mistake of believing one way or the other. Nonetheless, computers do give us a great chance to assist people in ways that other professions cannot. In addition, the study of computer science gives us a unique window into the mind of our Creator, allowing us to glorify Him by creating, and by serving; two actions that the Bible shows as integral to who God is. It is these two aspects of God’s nature that inspired the Choir Maker project, and have guided its progress so far.

First, creating. Before anything else in the Bible, the first thing we learn about God is that He created everything, that He is The Creator. A significant portion of the first book of the Bible is dedicated solely to the details of how, what, and when God created which things. It is made clear that He created, not on a whim, but because it is part of who He is. After each creation, we hear God say that what He created was good, or even very good, and through that we see Him enjoying the act of creation. He is an artist who creates, then steps back and rejoices because the piece of art He was working on is good. It stands to reason then, that as bearers of the image of God Himself, we too would take joy in creation. So, just as the painter has his canvas, or in some cases the roof of a cathedral, the programmer has his text editor. Just as the sculptor has a variety of tools to help him sculpt, the programmer has a variety of languages, libraries, IDEs, and the like to assist him. More explicitly, programming is an art in the same way that painting or sculpting is an art, and just as the artist paints beautiful paintings, or sculpts amazing sculptures, so the programmer can write truly beautiful programs. Donald Knuth explains this idea of programming as an art in one of his online publications:

When I speak about computer programming as an art, I am thinking primarily of it as an art *form*, in an aesthetic sense. The chief goal of my work as educator and author is to help people learn how to write *beautiful programs*. . . . My feeling is that when we prepare a program, it can be like composing poetry or music; as Andrei Ershov has said [9], programming can give us both intellectual and emotional satisfaction, because it is a real achievement to master complexity and to establish a system of consistent rules.  
 Furthermore when we read other people's programs, we can recognize some of them as genuine works of art. . . . The possibility of writing beautiful programs, even in assembly language, is what got me hooked on programming in the first place.  
 Some programs are elegant, some are exquisite, some are sparkling. My claim is that it is possible to write *grand* programs, *noble* programs, truly *magnificent* ones![[1]](#footnote-1)

So we see programming as an art? that programming is an art, and as with any art, we as Christians should find joy in programming in the same way that a sculptor finds joy in a beautiful sculpture, or our Creator found joy in creating us.

is that simply by writing programs we are, in some measure, imitating our Lord. As Scott Hoelsema so artfully puts it in his paper *Food Pantry Manager*:

God is the Creator, and humans, created in His image, cannot help but feel the drive to create. Though there are a number of outlets for this creative work, few so strikingly mirror God's acts of creation as computer programming. . . . Programming establishes not just the work, but the laws that govern it. Perhaps it would be fitting to say that programming, though faintly, is the best analogue we know to God's mode of creation – speaking.[[2]](#footnote-2)

Just like when we paint beautiful paintings, or write amazing stories, when we write a program, we are imitating our Creator’s creativity.

Now that we have seen how creation is a vital part of our Creator’s nature, and that as programmers we have an amazing opportunity to imitate Him in this way, we should discuss another vital aspect of God that programmers have unique opportunities to imitate: service. Just as the Son of Man came to serve, not to be served, we are also called to serve others. It is as Jesus says, if He came to serve, and we are His servants, how much more are we to serve?

1. http://www.paulgraham.com/knuth.html [↑](#footnote-ref-1)
2. [↑](#footnote-ref-2)