**The Way: Computer Programming with Python**

**Chapter 1 Introduction**

Chapter Overview:

* Introduce Jesus through John the Baptist
* Believing in Christ will create in us a desire to be faithful to His commands
* Introduce programming and tools needed with python
* A desire to be good engineers creates a desire to follow best practices
* Introduce what will be taught in this book

Chapter Thesis: Faith in Jesus, believing God gave his only begotten son for our sins, results in a desire to be obedient to His commands. However, we don't focus just on the obedience to commands, we focus on God. Paul taught us that it is not our actions that save us but our faith. Having a desire to please the Almighty will want us to be obedient to His commands. As in engineering, we want to be great engineers and scientists. As such, we follow best practices and proven techniques. The sole focus is not just on best practices, we focus on wanting to be good engineers and scientists and as such we seek best practices. That is no different than wanting to be a great Christian. Although it is more rewarding in being a Christian.

**Introduction**

Computer programming equips our students with logical thinking and problem solving while preparing them for a world in which computing is increasingly pervasive. Computer science is no longer an elective **if and only if** our children would "like" to learn, it is a necessary skill for living in the world today. The same is True with God's will, it is no longer "nice" or an "elective" if kids would like to learn God's will, it is necessary to not only live in this fallen world but more importantly to obtain everlasting life in Heaven. Computer knowledge is important, but it is only a tool that we use to thrive in our Babylonian-time and deliver the Word.

In implementing God's Will, individuals discover that the use of computer science concepts and technologies can effectively aid in their advancing of God's messages. While there are initiatives for integrating programming into our K-12 curricula, the challenge is in preparing our educators to teach the discipline effectively at all grade levels. This book is an essential and practical guide for every K-12 teacher, home schooling parent, and first year undergraduate students. The users of this book will find 8 chapters that cover fundamental concepts of the python programming language as well as basic concepts of computer science all reenforced with practice problems and coding challenges.

The richness of this textbook lies in the fact that it fills a large chasm in not only teaching computer programming but also integrating biblical concepts into the teaching. It is challenging at times to figure out how particular biblical themes integrate into certain classroom topics, and it can be quite difficult with technology-based classes like computer science. To date there is no textbook that equips our educators and home schoolers with the necessary tools in teaching computer programming with Christianity as its framework. This book does just that and shows, not surprisingly, how God reveals Himself in the beauty of Scripture and in the 1s and 0s that allow us to instruct machines and use computer technology.

Computer programming may be new to a lot of people but reading scripture and learning about Jesus may be new to a lot of people as well. Where would you start in introducing someone to Christ? Where do you start in introducing computer programming? Are they even related? Is the person going to be scared or worried about what they may learn or discover about themself? Are they going to think that learning about God is way too complex or over their head?

|  |  |
| --- | --- |
| These questions may result in being ***True*** and are applicable when learning computer programming as well. Programming a computer may seem foreign to a lot of people but with faith we can do it. We need to have faith that if it is God's will then we will be able to accomplish whatever God has in store for us. | *Boolean values are data types used in computer programming that represent a truth value. In our programs we can use integer values, another data type that represents real numbers (i.e. -2352, +5674, etc.) or float data types (i.e. 3.1415, 2.718, etc.). Another data type is the Boolean data type which can take on values of True and False.* |

**Introduce Christ**

To start off any journey it is always good to start off with introductions. As such, put yourself in the year 27 AD and John the Baptist is preaching in the Jordan valley. John the Baptist is specifically preaching a message of repentance, where folks should make every effort to remove sin from their life and make it easy for the coming Christ to get to them. This was common in the day of traveling royalty, in which traveling kings and queens would send ahead a party to make sure that the road, or the way, was easily passable. If they needed to fill in valleys or straighten out paths, whatever was needed so that the king would be able to get to his people. John the Baptist is doing the same thing except he is trying to make the path to each person passable for the one true King, the Messiah. As such, John is preaching that people need to ask for forgiveness for their sins and they must stop sinning! John is using the analogy that sin represents valleys and mountains, and people must remove sin so that God can get to them.

|  |  |
| --- | --- |
| *When we have one event following another event, we say that these events are sequential to one another. Event A was John the Baptist and Event B was Jesus Christ. Event B had to follow Event A; this is considered sequential. If Event B could have happened at the same time as Event A then we would say these two events are parallel to each other. But we know that John the Baptist preceded the coming Christ, so these two events are considered sequential.* | John was also preparing the people for the fact that Christ would come after him, that he was not the Christ but the Messiah would be coming next. Now John the Baptist knew Jesus but only as cousins, he did not know or think of Jesus as the coming Messiah. John had said that he was told by the Holy Spirit that he would know the coming Christ when he baptizes Him and sees the Spirit come down from heaven as a dove and remain on him (John 1:32). When John baptized Jesus, this is exactly what he saw and he knew then that Jesus was the Son of God. |

John the Baptist was a teacher in the truest sense. He prepared the people of the Jordan Valley and equipped them for the coming Christ. Even though we had been equipped since the beginning of time, the world was now formally introduced to the Son of God. It is the apostle Paul that tells us in the book of Romans that Jesus didn't need any introductions. He was made known to us since the beginning. Paul argued that no one had any grounds to say that they did not know Christ. That it was already upon people's hearts and it was left to them to either accept or reject the Truth. In accepting the Truth Paul taught us that it was wrong to think that one could buy their way into Heaven and God did not hand out His grace because of what one does. One cannot simply go out and help an old lady cross the street and turn around and say "Okay God, I did my task, hand out your grace!" It does not work like that, God hands out His grace by faith in Him alone.

Accepting Jesus and having faith in God's hope is what earns the grace of God. Now in knowing that it does not mean you can run out and continue sinning and say "well, I have faith in God that He will forgive me so it is alright!". It doesn't work like that either, as we are taught in the book of Romans. It works by having faith in God, quite literal. Simply having faith in God is what will create in you a desire to live by God's commands. Consider this, you are not complying with God's commands simply because you **should** comply with God's commands. You comply with God's commands because you love God and have faith in Jesus. The love you have for God creates the desire in you to want to follow His commands. When you love someone, you don't want to hurt them. When you love someone, you want to do things they like, you want to live up to their expectations, you want to please them because you love them. You don't please them or do things they like SO that they like you, you do it because you love them.

This is an extremely important point and cannot be taken lightly. Following God's commands is a good thing, but you should follow God's commands because you love and fear God. Your desire to be a servant of God creates a behavior in us to want to be obedient. As in engineering, we want to be great engineers and scientists. As such, we follow best practices and proven approaches to help mature our engineering practices. This is no different than wanting to be a great Christian. Although it is more rewarding in being a Christian.

**Introduce Computer Programming**

In 2020 the web site Indeed had software architects, software developers, data scientists, IT Security Specialists as part of their top 10 jobs based on salary and growing abundance of opportunities. It is no doubt that engineering and science opportunities continue to grow exponentially in our Babylonian world. Like Daniel that was taken into Babylon, we need to understand how best we can serve Christ during these difficult times.

Financial stability, job security, intellectual stimulation are all good reasons to go into computer programming. But programming does something else that attracts and retains us. It fosters creativity, reasoning and problem solving. It provides an opportunity to create something from nothing, using logic to turn computer constructs into viable functioning solutions. Programming can be quite challenging, where we get to work on the cutting-edge of technology where there is never a dull moment. The biggest reward programming gives though is an opportunity to serve God using the talents and skills that He has given us. It is through prayer that we discover how God would like for us to apply these skills. We must always remember, we do not do things because they are easy, we do these things because they are hard.

|  |  |
| --- | --- |
| So why learn a programming language? To make sure we are all on the same page, we must use a computer programming language to instruct or tell the computer to do something. If I want the computer to add two numbers together then I need to tell the computer to do it using a language that it understands. | *x = a+b*  *The above is a programming statement that adds the value stored in variable a to the value stored in variable b and then stores the result of that addition in the variable x.* |

So why learn Python as your computer programming language? First of all Python is one of the easiest programming languages to understand. So right off the bat you won't be overwhelmed with a complex language. Second of all the user community is quite large for Python, which means that if you run into a problem a simple search on the World Wide Web (WWW) will provide numerous answers.

|  |  |
| --- | --- |
| *An Integrated Development Environment (IDE) is a software tool that you run on your computer and allows you to write and edit programs like a text editor. It also allows you to execute the programs that you wrote and it allows you to debug the programs when there are problems.* | Third, Python is easy to experiment with using an Integrated Development Environment (IDE). It does not take a whole lot of time and resources to try out some programs and get started. Fourth, in addition to having a large support community there is also a lot of libraries or modules that have already been written for Python. This means that it is easy to reuse code from others so that you don't have to build everything to just get up and running. |

Lastly, Python code runs anywhere that has a Python interpreter. Today, almost everything has a Python interpreter and so basically if you write your code on your computer then there is a good chance that it will run on your friend's computer and you can share programs easily.

So how do you go about learning the computer programming language? Just like reading the Bible. You have to read it and then you have to do it. You must know Scripture, so you read the Bible. But that is only part of God's solution. The other part is that you must **DO** what God wants you to do! Just knowing Scripture and being able to recite verses means nothing. Even the demons can do that. But if you behave as God wants you to behave, if you do what God wants you to do then you are demonstrating your faith in God. Well, computer programming is no different. You must read about the Python programming language and then you must **DO IT**. You should spend more time practicing the language than you do reading about. When reading a concept in this book I highly recommend that you not only try out the examples but you also explore the examples, by changing things around and experimenting with what happens.

The tools that you need to learn are a computer, Python and the Python **I**ntegrated **D**eve**L**opment **E**nvironment (IDLE). IDLE provides your syntax highlighting, text editing, and code execution all in one software package. You can download the software tool at [www.python.org](http://www.python.org)

Once installed you then open "IDLE Python X.X" application which will look like figure 1

|  |
| --- |
| Figure 1. When you first run IDLE Python this is the "shell" that starts up |

The shell is part of the development environment that allows you to run Python commands and allows you to execute your python program. The 3 greater-than signs (>>>) is called the prompt and that is where you type in Python commands. This is one way of writing Python programs, it not the most efficient way and we will change later in the book and do it differently, but it is easy way for us to get started.

For example, at the prompt type:

*print("Hello God's Creation")*

|  |  |
| --- | --- |
| And hit Enter. After hitting Enter you should see the string "Hello God's Creation" printed in the terminal window below the command that you entered (see Figure 2). | *A string is just another data type like integer, Boolean, and float. A string is represented by the quotation marks (“) and the string is the set of characters between quotation marks. For example, if you see “hello” then the string is the set of characters hello.* |

|  |
| --- |
| Figure 2. The results of entering in your first Python command |

The word ‘print’ is part of the Python language and you are telling the computer to display to the screen whatever you have typed within the quotations, “Hello God's Creation”. The list of characters that we type within quotation marks is called a String and that is a data type like a boolean value or an integer value (we will discuss these later).

Figure 3 breaks this command down so that we can discuss its parts.

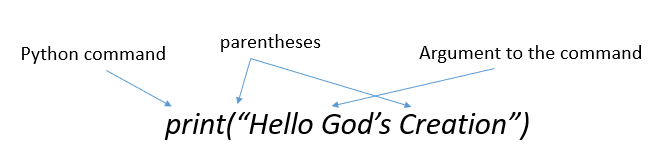


Figure 3. The print command broken down

The parentheses tell the computer that print performs a function and can take an argument. The argument that you are passing to the Python function print, is the string “Hello God's Creation”.

This line of code represents a Python executable statement to display something to the screen. Now you can mess around with that line of code and explore other effects like changing the string to be printed to the screen to something like your name and see the results. Remember, in that one line of python executable code we have learned:

1. Python command called *print*
2. *print* is actually a function represented by the parentheticals (i.e., print())
3. The function *print* takes one argument, a string
4. The argument is a string because it is encapsulated with quotations

Now the last thing we need to introduce are the use of files while working with Python. What you just did above is work from the command line, specifically the interactive shell. That works and is beneficial while learning certain aspects of Python. But eventually we will want to start saving our python code in a file and then run the whole file at one time versus one command at a time. An example workflow will be something like that in figure 4.

|  |
| --- |
| Figure 4. Basic programming workflow for writing code in a file, save the file, and then execute the file (i.e., run the file of python commands sequentially) |

To create a Python file you do it from the interactive shell:

1. In the IDLE shell you choose *File* and then *New File* (see Figure 5) from the menu

|  |
| --- |
|  |

Figure 5. Creating a new Python file using menu bar, select *File* and then *New File*

1. An empty window will appear with "Untitled" in the menu bar
2. In the newly created file, you enter in the Python code to print your name to the screen

*print("Hello Marcus Fisher")*

1. Then choose *File* and then *Save* on the menu bar and give your file a name
   1. Any name will work, something like *"intro.py"*
   2. All python files end with the extension *.py*
2. Then choose *Run* on the menu bar and then *Run Module*
3. The python file or program will execute in the interactive shell and you will see the results of your program executing in the interactive shell (see Figure 5 above)

Make sure you follow along with these examples but also explore these concepts making sure that you can create another file, add several print statements to the file and see their results when you execute the file in the interactive shell.

Challenges and Exercises:

1. When creating a new file in Python the original file is empty until enter in your Python code. What else is empty until you fill it with Jesus Christ?
2. What does faith mean to you, discuss this topic from a Christianity perspective as well as a general perspective. In addition to writing about what faith means to you, where do you think your faith will be challenged?
3. When John the Baptist introduced Jesus to the World, what was the first question that John the Baptist's disciples asked Jesus? and why do you think that they asked Him that?
4. What does an Integrated Development Environment do for you?
5. When you see a Python statement that has parentheticals, then what is that python statement called? As an example, what is this python statement also known to be: *input()*