

Chapter

5

Shell Script Programming

In this chapter, you will learn the following to World Class standards:

- 1. Shell Scripts**
- 2. Variables**
- 3. Shell Script Operations**
- 4. Creating an Executable Shell Script**

Introduction to Shell Scripts

A shell script is a text file that has one or multiple lines of programming language and UNIX commands. The purpose of a shell script is to accomplish a small task automatically when the script is executed. There are three key skills you need to learn to successfully execute a shell script: how to create the file in a Linux or UNIX editor, how to make the file executable, and how to run the file.

```
let a=3
let b=7
let c=16
let d=a+b*c
echo $d
```

Figure 5.1 – A Very Simple Shell Script

Variables

There are different types of variables we can use in a shell script. Some are numbers and others are text strings. To create a variable, type the name of the variable, an = sign, and then the value of the variable. A variable name can only be one word (no spaces) so if you wish to include two words, separate them with an underscore. To create a text string variable, put the text in quotations after the =. Once we create a variable, we can see its quantity by typing **echo \$** and then the variable name. The system will respond by displaying the value of the variable. Usually we will want to program in all lower-case text because the smaller character makes it easier to troubleshoot large, dense quantities of code.

We can add other information to variables as well. For example, setting **'date'** to a variable titled **now** will display the current date when we echo **now**. To show just the month – day – year, type **now = `date` + "%m-%d-%y"**

We can set the current path as a variable as well by typing **path=\$PATH**. This will save us a lot of time and code from not having to enter long directory paths.

```
answer="The answer is"
a=3
b=7
c=16

echo $answer
The answer is

Echo $a
3
```

Figure 5.2 – Creating and Recalling Variables

```
now=`date`

echo $now
Wednesday 06/01/10
```

Figure 5.3 – Creating a Date Variable

```
path=$PATH
echo $path
```

Figure 5.4 – Creating a Path Variable

Simple Shell Script Operations

The most basic programming operations we can compute in shell scripts are basic math calculations. Adding is a simple task in shell scripting. First we set the variable to a letter or name and then we use the plus + operator to add two or more variables. We can also add regular numbers in the equations.

In our example shown, the variable **a** is added to the variable **b** and set to the variable **c**. When we echo the variable **c**, we see the answer 10.

Subtracting is also an easy task in Shell Scripting. We set the variable to a letter or name and then we use the minus sign - to subtract one or more variables. We can also subtract regular numbers in the equations.

In our example shown, the variable **b** is subtracted from the variable **a** and set to equal the variable **c**. When we echo the variable **c**, we see the answer -4.

The next operation we will cover is multiplying in a shell script. We set the variable to a letter or name and then we use the asterisk * to multiply one or more variables. We can also multiply regular numbers in the equations.

In our example shown, the variable **a** is multiplied by the variable **b** and set to the variable **c** with the equal sign. When we echo the variable **c**, we see the answer 21.

```
a=3
b=7
c=a+b

echo $c
10
```

Figure 5.4 – Simple Addition in a Shell Script

```
a=3
b=7
c=a-b

echo $c
-4
```

Figure 5.5 – Simple Subtraction in a Shell Script

```
a=3
b=7
c=a*b

echo $c
21
```

Figure 5.6 – Simple Multiplication in a Shell Script

The last operation we will cover is division in a shell script. We set the variable to a letter or name and then we use the forward slash / to divide one or more variables. We can also divide regular numbers in the equations.

In our example shown, the variable **a** is divided by the variable **b** and set to the variable **c** with the equal sign. When we echo the variable **c**, we see the answer 0.42857142.

The mathematical Order of Operations is followed in Shell Scripting. Operations in side parentheses are calculated first, then exponents, then multiplication and division from left to right and finally addition and subtraction from left to right.

In our example shown, the variable **b** is multiplied by the variable **c** which equals 112. Then the variable **a** is added to total 115. When we echo the variable **d**, we see the answer 115. A common math mistake for algebra beginners would be to find 160 as the answer, but this is incorrect.

```
a=3
b=7
c=a/b

echo $c
0.42857142
```

Figure 5.7 – Simple Division in a Shell Script

```
a=3
b=7
c=16
d=a+b*c

echo $d
115
```

Figure 5.8 – Order of Operations in a Shell Script

Creating an Executable Shell Script

To apply the programming skills we've covered so far, we can make an executable shell file to complete the operation we just used as an example for order of operations at the end of the previous section. Open the VI editor and write the example and write the following program

```
let a=3
let b=7
let c=16
let d=a+b*c
echo $d
```

Figure 5.9 – Code to Enter in the VI Editor

Save the program and make it executable by typing **chmod ugo+x calc**. Now run the program. The answer is 115.

*** World Class CAD Challenge 44-5 * - Write a Script that displays two message boxes, the first will contain the script name, copyright date and author. The second message will display information from the computer.**

Continue this drill four times using some other messages, each time completing the VBScript in less than 30 minutes to maintain your World Class ranking.