14.2 Shell Environments, Bash Variables and Parameters

As you study this section, answer the following questions:

- What is the purpose of the shell declaration (the shebang)?
- How can you comment out lines in a shell script?
- Which methods can you use to run a script from the command line?
- Why might you need to use the **declare** command in a script?

In this lesson, you will learn to:

- Use variables and parameters in a bash script.
- Use shell arithmetic in a bash script.
- Use arrays, variable expansions, and command substitution in a bash script.

Key terms for this section include the following:

Term	Definition
Environment variable	A bash variable that is inherited by child shells.
Shell variable	A bash variable usually added by a shell startup script.
Interactive shell	A shell that a user can interact with through stdin (the keyboard) and stdout (the display screen) that runs the bash startup scripts.
Login shell	An interactive shell that requires the user to enter a user ID and password.
Non-login shell	An interactive shell that doesn't require the user to enter a user ID and password.
Non-interactive shell	A shell that does not interact with the user like the shell that runs a script.
Positional parameter	A series of special variables that contain the arguments given when the shell is invoked or when a script is run.
Special parameters	A set of special variables that are maintained by the shell and contain values such as the number of positional parameters, the status of the last executed command, etc.
Integer variable	A variable that can be assigned a value based on the results of a shell arithmetic equation.
Shell arithmetic	A feature of the bash shell that evaluates arithmetic expressions.
Array variable	A bash variable that can hold multiple values that are referenced by the variable name and an index.

Indexed array	A bash array that is indexed using a number.
Associative array	A bash array where the index is a character identifier instead of a number.
Variable expansion	A bash construct in the form of ${} that substitutes the variables value for the name of a variable.
Command substitution	A bash construct in the form of $() or ` ` that substitutes the results of a command specified in the construct.

This section helps you prepare for the following certification exam objectives:

Exam	Objective
CompTIA Linux+	5.1 Given a scenario, deploy and execute basic BASH scripts. ● Shell environments and shell variables ○ export ○ env ○ set ○ printenv ○ echo ● Shell expansions ○ ${} ○ $() ○ ` ` ● Positional parameters