

# 14.1.5 Practice Questions

**Candidate:** Ethan Bonavida (suborange)

**Date:** 12/8/2022 8:56:47 pm • **Time Spent:** 00:39

**Score: 100%**

Passing Score: 80%



## ▼ Question 1: ✓ Correct

When creating a bash script, it is important to document the purpose of the script. Which of the following is a valid comment?

- ☐ // Comment text
- ☐ \$ Comment text
- ☒ # Comment text
- ☐ !! Comment text



### Explanation

Comments begin with a number sign (#). The shell ignores these lines when running the script. Comments help communicate how the script was constructed and what it is designed to do.

// will return the error "Is a directory."

\$ and !! will both return the error "Command not found."

### References

-  14.1.1 Bash Scripting Overview
-  14.1.4 Scripting Facts

q\_script\_lp5\_comment.question.fex

## ▼ Question 2:

✓ Correct

Troy, a system administrator, created a script to automate some daily administrative tasks. Which of the following commands would make Troy's script, /scripts/dailytasks, executable by everyone, but writable only by its owner?

- ➡ ☒ **chmod u=rwx,go=rx /scripts/dailytasks**
- ☐ **chmod 775 /scripts/dailytasks**
- ☐ **chmod u=x,g=x /scripts/dailytasks**
- ☐ **chmod 577 /scripts/dailytasks**

## Explanation

**chmod u=rwx,go=rx /scripts/dailytasks** sets the permissions for the owner to be able to read, write, and execute the script. Both group and other are assigned read and execute permissions.

**chmod 577 /scripts/dailytasks** would not give owner write permissions, but would give group and other write permissions.

**chmod u=x,g=x /scripts/dailytasks** would only give execute permissions to owner and group.

**chmod 775 /scripts/dailytasks** would give write permissions to group.

## References

 14.1.4 Scripting Facts

q\_script\_lp5\_script\_permissions.question.fex

▼ **Question 3:**

✓ Correct

Which of the following shell declarations should be entered on the first line of a script for a system that uses the bash shell?

- ☐ /bin/bash
- ☐ #!/bin/csh
-  ☒ #!/bin/bash
- ☐ /bin/tsh

**Explanation**

#!/bin/bash is the shell declaration that is added to the first line of a bash script. #! is referred to as a shebang or hashbang and is followed by the path to the shell.

/bin/bash is the path to the shell and is not the correct syntax for a shell script.

/bin/tsh is the path to the trusted shell, tsh.

#!/bin/csh would be used if the C shell was being used instead of the bash shell.

**References**

 **14.1.1 Bash Scripting Overview**

 **14.1.4 Scripting Facts**

q\_script\_lp5\_shell\_declaration.question.fex

## ▼ Question 4:

✓ Correct

From the bash command prompt, which of the following commands directly executes /usr/bin/scripts/cleanup.sh?

☐ **export /usr/bin/scripts/cleanup.sh**

☐ **cleanup.sh**

☐ **exec cleanup.sh**

➡ ☒ **source /usr/bin/scripts/cleanup.sh**

## Explanation

**source /usr/bin/scripts/cleanup.sh** directly executes the script.


**export /usr/bin/scripts/cleanup.sh** returns a "Not a valid identifier" error.

**cleanup.sh** returns a "Command not found..." error.

**exec cleanup.sh** returns a "Not found" error.

## References

 14.1.2 Bash Script Execution

 14.1.3 Executing and Sourcing a Script

 14.1.4 Scripting Facts

q\_script\_lp5\_source.question.fex

## ▼ Question 5:

✓ Correct












Which of the following are valid ways to assign a variable a value in a bash script? (Choose TWO.)

- ☐ **num1 := 7;**
- ➔ ☒ **declare -i num1=4**
- ➔ ☒ **variable1=Hello**
- ☐ **num1==5**
- ☐ **type string variable1=Hello**

## Explanation

**variable1=Hello** and **declare -i num1=4** are both ways to assign a variable a value. Declare is used to type a variable as an integer (whole numbers only). Variables hold values that the script uses when running. These values can be either numbers or text.

## References

-  2.5.1 Environment Variables
-  2.5.2 Manage Environment Variables
-  2.5.3 Environment Variable Facts
-  14.1.1 Bash Scripting Overview
-  14.1.2 Bash Script Execution
-  14.2.1 Bash Shell Environments and Shell Variables
-  14.2.2 Bash Shell Parameters, User Variables and Expansions
-  14.2.3 Bash Shell Variables and Parameters
-  14.2.4 User Variables and Shell Arithmetic
-  14.2.5 Arrays and Expansions
-  14.2.6 Shell Environments, Bash Variables and Parameters Facts

q\_script\_lp5\_variables.question.fex

**Copyright © 2022 TestOut Corporation All rights reserved.**