Bash Scripting

Quiz, 5 questions

5/5 points (100%)

✓ Congratulations! You passed!

Next Item

1/1 point

1. Which of the following commands are equivalent (Select all answers that apply)?

☑ **[[-f file.c]] && cat file.c**

Correct
You can use such continuations instead of explicit **if then else** constructs

☑ **if [[-f file.c]] ; then cat file.c ; fi**

Correct
The double brackets are in **bash**, but not **sh**

☑ **if [-f file.c] ; then cat file.c ; fi**

Correct
This is standard **bash** syntax with single brackets

☑ **if test -f file.c ; then cat file.c ; fi**

Correct
test is an older construct, probably best not used in modern scripts

1/1 point

2. Which commands will list all files under the current directory ending in "~" (Select all answers that apply)?

Note: Output lists may differ in format.

☑ **find . -name "*~" -ls**

Correct
This uses the **-ls** option to **find**

☑ **find . -name "*~" | xargs ls -l**

Correct
This shows the use of **xargs**

☑ **ls -l $(find . -name "*~")**

Correct
This substitutes the **find** command results into **ls** as arguments

☑ **find . -name "*~" -exec ls -l {} ';'**

Correct
This uses the explicit **-exec** argument to **find**

1/1 point

3. Functions (subprograms) are useful in **bash** scripts because (Select all answers that apply):

☑ They eliminate the need to retype the same set of commands more than once

Correct
This makes things shorter and reduces maintenance as changes are made and helps avoid errors

☑ They make things easier to read and comprehend

Correct
Shorter and less repetitive is always beneficial for comprehension

☑ It is better not to have to call another script to get things done

Correct
This requires keeping track of more than one file, making sure they are all in the path, etc.

1/1 point

4. How would you get the value of a variable named **VAR** into a script?

● **read VAR**

Correct
This will read the variable in and then its value can be used as **$VAR**

○ **input VAR**

○ **ask VAR**

○ **accept VAR**

1/1 point

5. Select the correct statement:

○ A **bash** function can be placed anywhere in a script, before or after it is used

● A **bash** function must be placed before it is used in a script

Correct
bash scripts are not compiled, just interpreted sequentially