1. What is a shell script?  
a) group of commands  
b) a file containing special symbols  
c) a file containing a series of commands  
d) group of functions

Answer: c  
Explanation: When we have to execute a series of commands altogether, we store them in a file which is itself executed as a shell script. A shell script is basically a computer program designed to be run by the UNIX shell.

2. Shell scripts need to be saved with an extension .sh .  
a) True  
b) False

Answer: b  
Explanation: It’s not mandatory to save script files with .sh extension but we do so for our own convention as it makes it easy to match them with wildcards.

3. Shell scripts are executed in a separate child shell process.  
a) True  
b) False

Answer: a  
Explanation: Shell scripts are executed in a separate child process, and this sub-shell need not be of the same type as our login shell. In other words, even if our login shell is bourne, we can use a Korn sub-shell to run our script.

4. The first line in any shell script begins with a \_\_\_\_\_  
a) &  
b) !  
c) $  
d) #

Answer: d  
Explanation: When the comment character (#) is placed anywhere in a line; the shell ignores all characters on its right. However, this rule doesn’t apply to the first line which is the interpreter line. It always begins with #! and followed by the pathname of the shell to be used for running the script.

#!/bin/sh // first line defining the pathname

# script.sh // name of the script

5. To run the script, we should make it executable first by using \_\_\_\_\_  
a) chmod +x  
b) chmod +r  
c) chmod +w  
d) chmod +rwx

Answer: a  
Explanation: Before we run the script, it is essential to make the script executable first. After that invoke the script name to run the script. For making the script executable, we have to use chmod +x script\_name.

6. To spawn a child of our own choice for running the script, we can use \_\_\_ command.  
a) ps  
b) pr  
c) sh  
d) $$

Answer: c  
Explanation: We know that shell scripts are executed by a child shell. But we can also explicitly spawn a child of our own choice by using the sh command along with script name as an argument. When used in this way, the interpreter line is ignored by the shell.

7. Which command is used for making the scripts interactive?  
a) ip  
b) input  
c) read  
d) write

Answer: c  
Explanation: read command is used for making scripts interactive. It is used for taking input from the user. Input supplied from the keyboard is entered into the variable used with the read command. For example,

#!/bin/sh

# emp.sh

echo ”enter your name”

read Uname //read input from the user

echo $Uname // display input entered by the user

8. read command is shell’s internal tool.  
a) True  
b) False

Answer: a  
Explanation: read command is the shell’s internal tool for taking input from the user i.e. it makes the scripts interactive.

9. A single read statement can be used with one or more variables.  
a) True  
b) False

Answer: a  
Explanation: A single read statement can be used with one or more variables to let us enter multiple arguments. For example,  
**read FirstName LastName**

10. What are positional parameters?  
a) special variables for assigning arguments from the command line  
b) pattern matching parameters  
c) special variables for reading user input  
d) special variables and patterns

Answer: a  
Explanation: Shell scripts can also take input from command line. When arguments are specified with a shell script, they are assigned to certain special variables called positional parameters.

11. The first argument is read by the shell into the parameter \_\_\_  
a) 1$  
b) $3  
c) $$  
d) $1

Answer: d  
Explanation: Command line arguments are stored into positional parameters. The first argument is read by the shell into the parameter $1, second argument in $2 and so on.

12. The complete set of positional parameters is stored in \_\_\_\_\_\_ as a single string.  
a) $n  
b) $#  
c) $\*  
d) $$

Answer: c  
Explanation: There are some special parameters used by the shell. One of which is $\*, which stores the complete set of positional parameters as a single string.

13. Which of the following is used for storing the number of positional parameters?  
a) $n  
b) $#  
c) $\*  
d) $2

Answer: b  
Explanation: $# is used for storing the number of arguments specified. It lets us design scripts that check whether the right number of arguments have been entered. For example, the following script shows the use of positional parameters:

#!/bin/sh

# emp.sh

echo “ The number of arguments specified are $#

The arguments are $\*”

Now execute the script:

$ emp.sh director.lst // director.lst is command line argument given to the script.

The number of arguments specified is 1

The arguments are director.lst