Bash Shell Scripting – Solutions

Lab 1: Exit Status Code Usage

```
#!/bin/bash
# exitLab
#
# example of exit status
# check for non-existent file
# exit status will be 2
# create file and check for it
# exit status will be 0
#
ls xyzzy.345 > /dev/null 2>&1
status=‘echo $?‘
echo "status is $status"

# create the file and check again
# status will not be 0
touch xyzzy.345

ls xyzzy.345 > /dev/null 2>&1
status=‘echo $?‘
echo "status is $status"

# remove the file
rm xyzzy.345
```

Lab 2: Working with Files

```
#!/bin/bash

# filesLab -
# demos several simple commands to create a directory,
# cd to that directory, echo ‘pwd‘
# create several files,
# put data into the files and list the files
```

```
echo "Welcome to File Creator"
echo "What name do you want to give to the new directory?"

# Get the new name from the user
read dirName

# make the directory
mkdir $dirName

# change the current working directory to the new directory
cd $dirName

# announce where we are
echo "This directory is called ‘pwd‘"

# create some files
touch file1 file2 file3

# put content into the files, the directory and file name
echo "This is $dirName/file1" > file1
echo "This is $dirName/file2" > file2
echo "This is $dirName/file3" > file3

# announce the file names
echo "The files in $dirName are: "
ls -hl

# show the content of the files
echo "The content of the files are: "
cat file1
cat file2
cat file3

echo "Goodbye"
```

Lab 3: Environment Variables

```
#!/bin/bash

# ifLab demonstrate the use of environmental variables
# and the if-then-else clause
# Does not check to see if the user puts in i
# something other than a number
# It takes a number (1 or 2) then sets the
# variable MYANS to yes or no
# If 1 or 2 is not entered, the variable is set to unknown
# declare the variables
no="No" # 1
yes="Yes" # 2
unknown="Unknown" # default

# set a default value
dValue=$unknown

echo""
echo "This program accepts a number used to set the
environmental"
echo "variable MYANS to yes (1) or no (2)."
echo ""
echo "Enter the number 1 or 2:"
# this variable contains the number input by the user
read aValue

# check to see if the user input a value
# set the value to the user's input, if there is one
# otherwise set it the a default value

if [ $aValue -eq 1 ]
then
    MYANS=$yes
else
    if [ $aValue -eq 2 ]
    then
       MYANS=$no
    else
       MYANS=$dValue
    fi
fi

export MYANS
echo "The value of MYANS is: $MYANS"
```

Lab 4: Functions

```
#!/bin/bash
# functionLab
# demonstrates functions and script parameters
#
# Functions

func1() {

echo " This message is from function 1"

}

#-----------------------------
func2() {

echo " This message is from function 2"

}

#-----------------------------
func3() {

echo " This message is from function 3"

}

# Main script

# prompt the user
echo "Enter a number from 1 to 3"

# get the user's choice
read choice

# the number chosen by the user is added to
# the name func to select
# func1, func2 or func3
# the function call is simply the name
# of the function

func$choice
```

```
# put out a line feed
echo ""
```

Lab 5: Arithmetic

```
#!/bin/bash
#
# arithmeticLab
# demonstrates arithmetic, functions and simple if clauses
# three methods are used for arithmetic.
# the exercise requires only one.
# the three methods are:
# 1) let
# 2) expr
# 3) $((...)
# The user will input a letter and two numbers.
# the letter will
# be a(dd), s(subtract), m(ultiply) or d(ivide)
# to select an
# arithmetic operation.

# Functions. must be before the main part of the script
#
adder() {

# method 1. use let
let answer1=($fNumber + $sNumber)

# method 2. use expr
answer2=‘expr $fNumber + $sNumber‘

# method 3 use $((...)
answer3=$(($fNumber + $sNumber))

} # end adder function
#---------------------------

subtracter() {

# method 1. use let
let answer1=($fNumber - $sNumber)

# method 2. use expr
answer2=‘expr $fNumber - $sNumber‘
```

```
# method 3 use $((...)
answer3=$(($fNumber - $sNumber))

} # end subtracter function

#---------------------------

multiplyer () {

# method 1. use let
let answer1=($fNumber * $sNumber)

# method 2. use expr
answer2=‘expr $fNumber \* $sNumber‘

# method 3 use $((...)
answer3=$(($fNumber * $sNumber))

} # end multiplyer function

#---------------------------
divider() {

# method 1. use let
let answer1=($fNumber / $sNumber)

# method 2. use expr
answer2=‘expr $fNumber / $sNumber‘

# method 3 use $((...)
answer3=$(($fNumber / $sNumber))

} # end divider function
# End of functions
#

# Main part of the script
# check that user provided a letter and two numbers
# does not check to see if the user put in
# an incorrect letter
# which will simply display messages without an answer

if [ $# -lt 3 ]
then
     echo ""
     echo "Usage: Provide an operation (a,s,m,d) and two
     numbers"
     echo ""
     exit 1
fi

#-----------------------------------------------------------
-
# set the input numbers to variables to pass to the
functions
#

fNumber=$2
sNumber=$3

if [ $1 == "a" ]
     then
          adder
fi

if [ $1 == "s" ]
then
subtracter
fi

if [ $1 == "m" ]
then
     multiplyer
fi

if [ $1 == "d" ]
then
     divider
fi

#------------------------------------------------
# Present the answers for all three methods
#

echo "Method 1 Answer is $answer1"
echo "Method 2 Answer is $answer2"
echo "Method 3 Answer is $answer3"
```