

## INFORMATION TECHNOLOGY EDUCATION DEPARTMENT

**IOS 102 LAB**

**(Operating Systems Laboratory)**

**<LABORATORY SCHEDULE>**

EXERCISE

9

**Linux Shell Scripts using Conditional Statements**

<STUDENT NAME 1>

<STUDENT NAME 2>

DATE

**Experiment No. 9: Linux Shell Scripts using Conditional Statements**

**Objectives:**

In this experiment, the students are expected:

* to create shell scripts using the conditional statements such as if-else and switch-case statements

**Discussion:**

**Condition is defined as:**

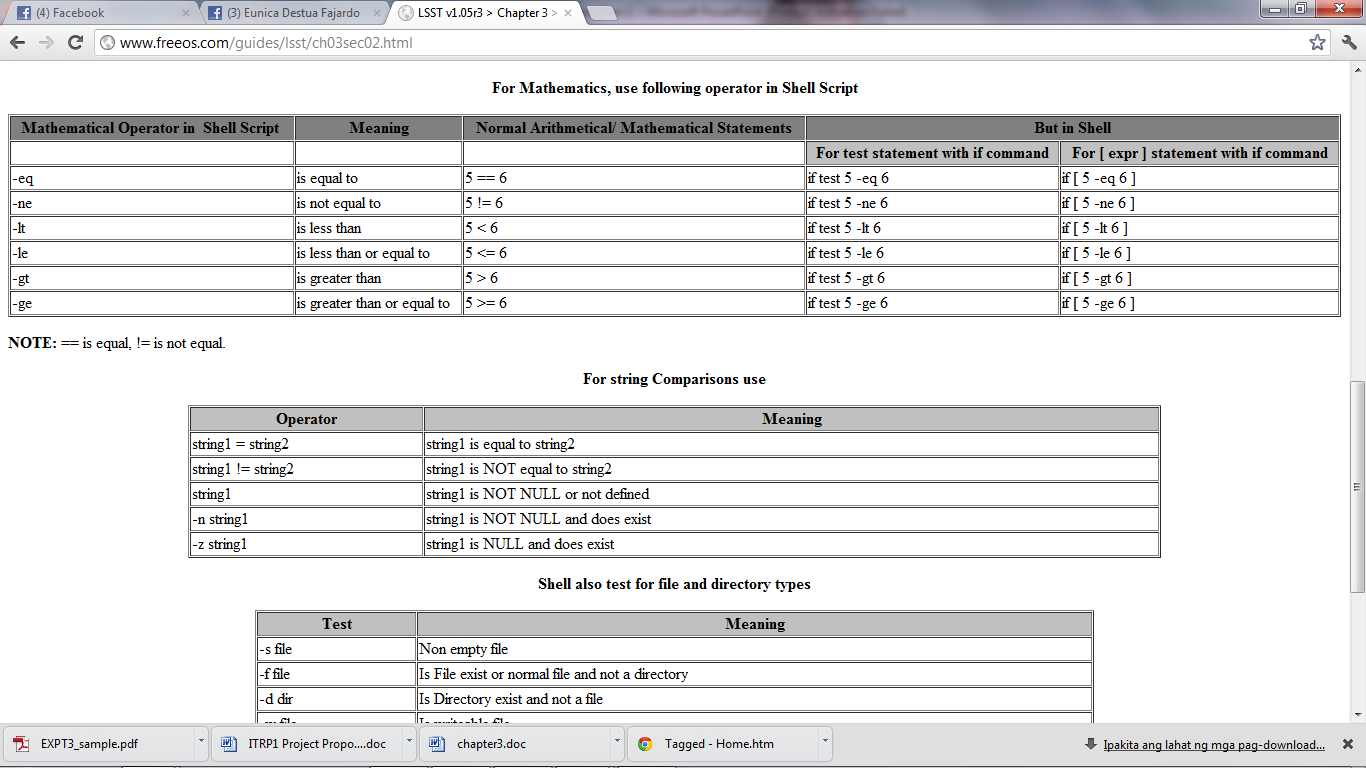
"Condition is nothing but comparison between two values."

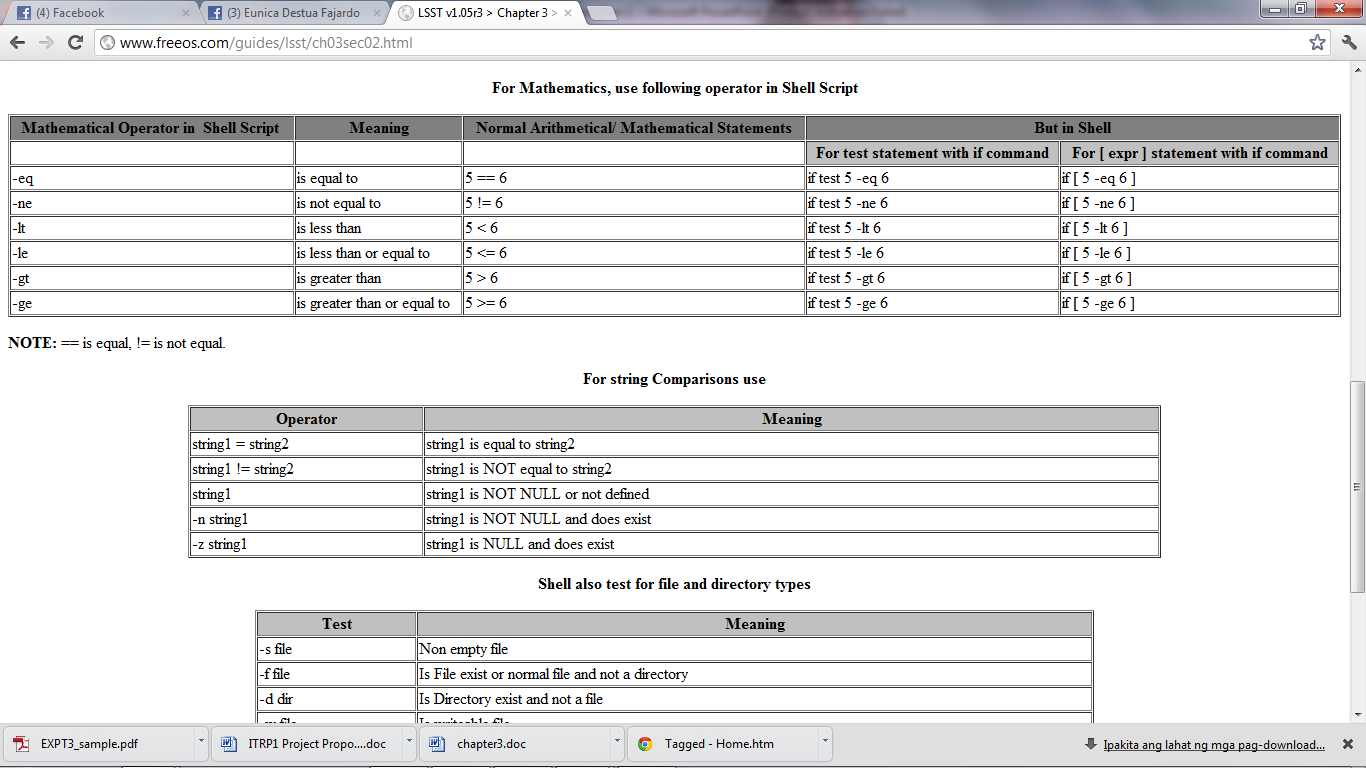
**Expression is defined as:**

"An expression is nothing but combination of values, relational operator **(such as >,<, <> etc)** and mathematical operators **(such as +, -, / etc )."**

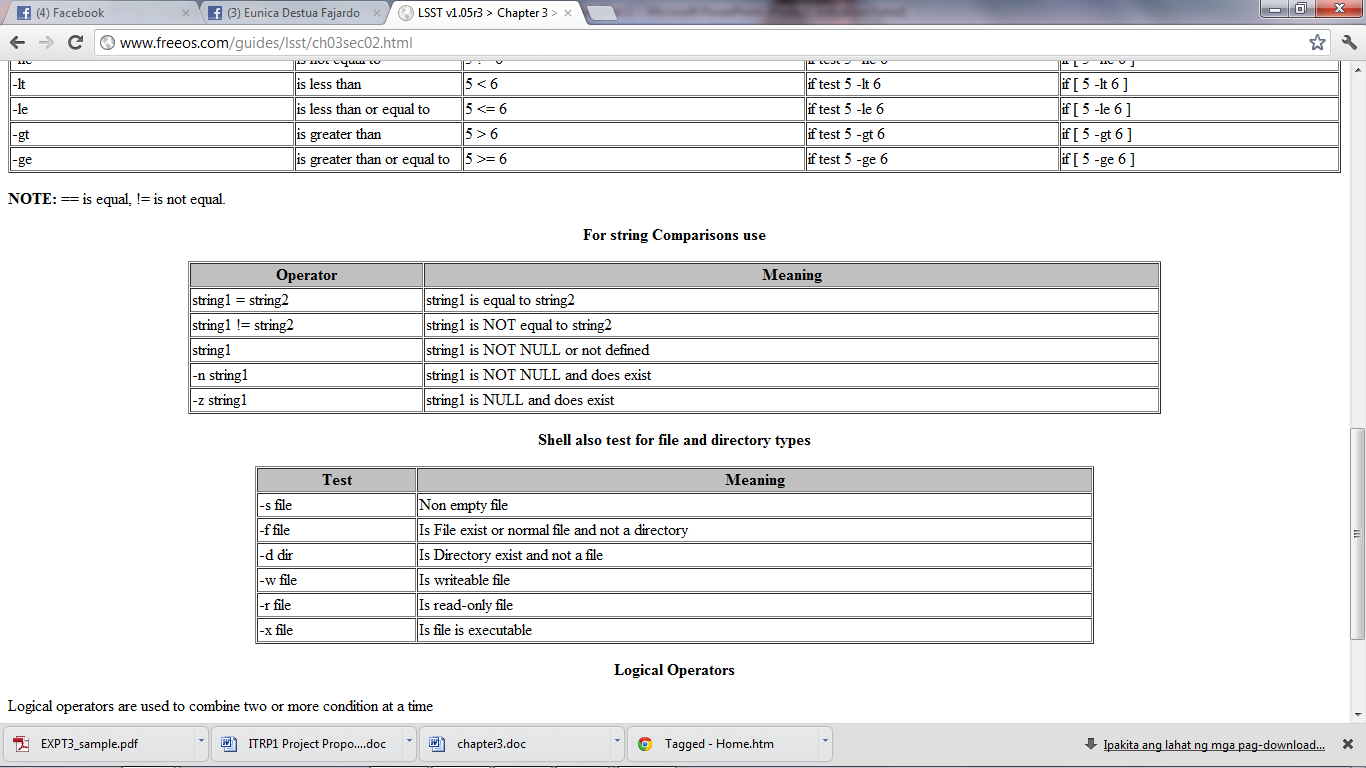
**Condition Statements**

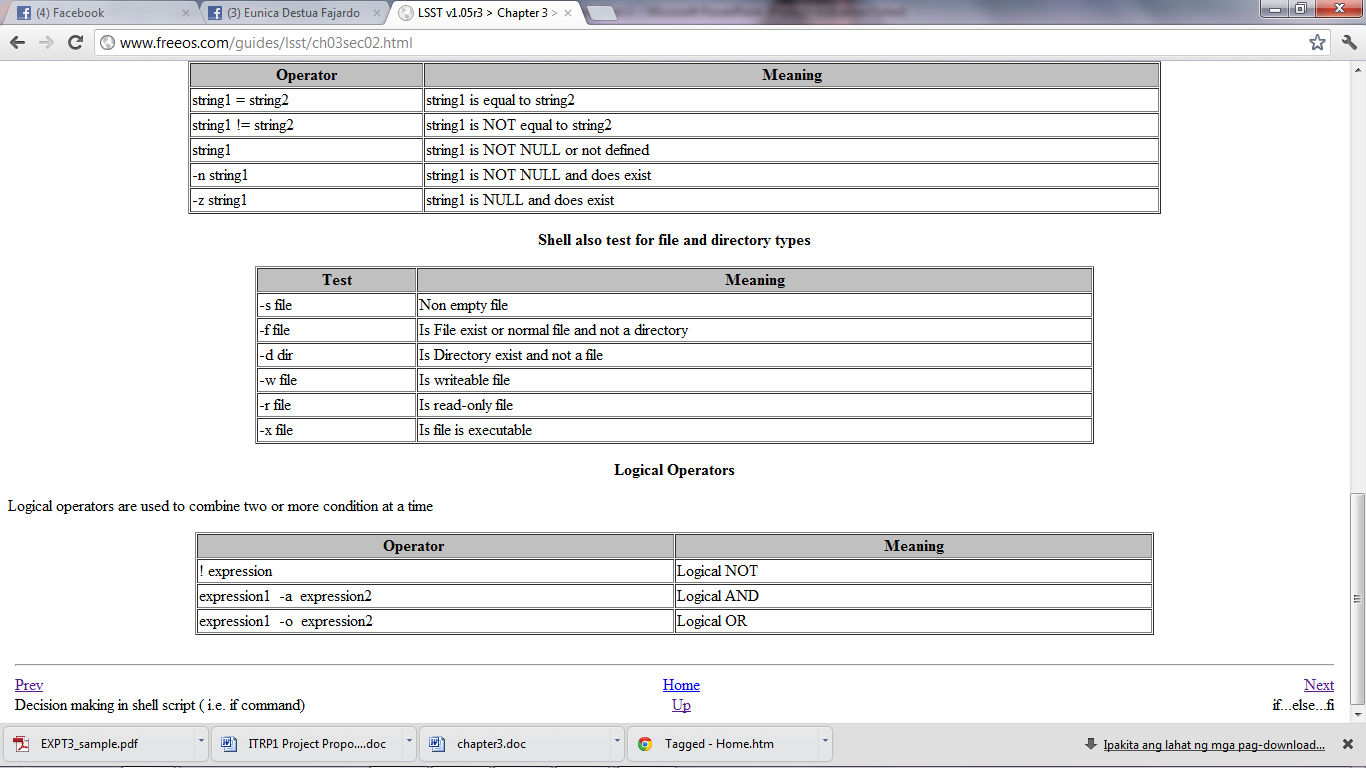
* **Relational Operators**





* **String Comparison and Logical Operators**





* **if condition**
* **if condition** is used for decision making in shell script, if given condition is true then command1 is executed.

*Syntax:*

if condition

then

command1 if condition is true or if exit status

of condition is 0 (zero)

...

...

fi

**Condition Statements**

* **test command or [ expr ]**

test command or [ expr ] is used to see if an expression is true, and if it is true it return zero(0), otherwise returns nonzero for false.

*Syntax:*

**if test expression**

*Example:*

**if test $x –eq 0**

*Syntax:*

**if test [ expr ]**

*Example:*

**if [ $x –gt 0 ]**

* **Case Statements**

The case statement is good alternative to Multilevel if-then-else-fi statement. It enables you to match several values against one variable. Its easier to read and write.

*Syntax:*

**case $variable-name in**

**pattern1) command**

**...**

**command;;**

**pattern2) command**

**...**

**command;;**

**patternN) command**

**...**

**command;;**

**\*) command**

**...**

**command;;**

**esac**

**Exercises:**

**1. Using test command**

Following script determine whether given argument number is positive**.**

**$ cat > pos1**

**# Script to see whether argument is positive**

**echo “Enter a number: “**

**read x**

**if test $x –gt 0**

**then**

**echo "$x number is positive“**

**else**

**echo "$x number is negative“**

**fi**

Paste your captured output below:

**2. Using [expr]**

**$ cat > pos2**

**echo “Enter a number: “**

**read x**

**if [ $x –gt 0 ]**

**then**

**echo "$x number is positive“**

**else**

**echo "$x number is negative“**

**fi**

Paste your captured output below:

**3. Multilevel (if…then…else)**

**$ cat > elf**

**echo “Enter a number: “**

**read x**

**if [ $x -gt 0 ]; then**

**echo "$x is positive"**

**elif [ $x -lt 0 ]**

**then**

**echo "$x is negative"**

**elif [ $x -eq 0 ]**

**then**

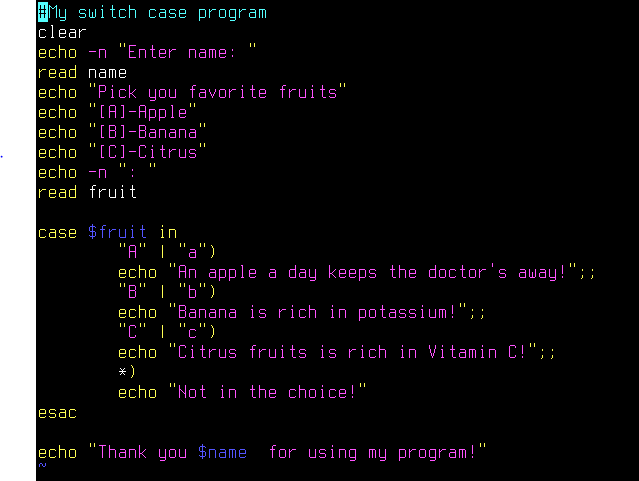
**echo "$x is zero"**

**else**

**echo "Opps! $x cannot be determined!"**

**fi**

Paste your captured output below:

4.

Paste your captured output below:

**Supplementary Problems:**

* Write a shell script that will perform/display the given outputs below:

**Screen 1:**

Welcome to Linux Scripts

Username:

Password:

Access Granted!

**Screen 2: (if access granted)**

Choose transaction

[a] – Grades Remarks

[b] – About

[c] - Exit

**Screen 3: (if access denied)**

Goodbye, “username”!

**Screen 4: (if “a” is chosen)**

**Enter grade:**

***Note:* Display the equivalent Grade Remarks of the input grade. Below is the Grade Remarks table:**

**100 – 90 Excellent**

**89 – 85 Very Good**

**84 – 80 Good**

**79 – 70 Fair**

**Below 70-0 Poor**

**Beyond 100 Invalid**

**Below 0 Invalid**

**Screen 5: (if “b” is chosen)**

***Note:* Display the programmers’ name**

**Screen 6: (if “c” is chosen)**

**Goodbye, “username”!**

**Answers:**

|  |
| --- |
| **Codes**  (Paste your captured codes below) |
|  |

|  |
| --- |
| **Sample Output**  Paste your captured output/s below. |
|  |