

Laboratory Assignments 3
Subject: Design Principles of Operating Systems
Subject code: CSE 3249

Assignment 3: Shell Programming using user defined variables, arithmetic operators, conditional statements.

1. Write a shell script **iaop** to perform integer arithmetic on two numbers, where the value of the two numbers will be given during runtime.

```
(munu@kali)-[~/Dos/assgn1]
$ nano iaop.sh

(munu@kali)-[~/Dos/assgn1]
$ chmod +x iaop.sh

(munu@kali)-[~/Dos/assgn1]
$ ./iaop.sh
enter two numbers
5
6
Sum: 11
Difference: -1
Product: 30
Quotient: 0
Remainder: 5
```

```
GNU nano 8.1 iaop.sh *
echo "enter two numbers"
read num1
read num2
sum=$(( $num1 + $num2 ))
echo "Sum: $sum"
diff=$(( $num1 - $num2 ))
echo "Difference: $diff"
prod=$(( $num1 * $num2 ))
echo "Product: $prod"
quo=$(( $num1 / $num2 ))
echo "Quotient: $quo"
rem=$(( $num1 % $num2 ))
echo "Remainder: $rem"
```

2. Write a shell script **faop** to perform floating point arithmetic on two numbers, where the value of the two numbers will be given during runtime.

```

(munu@kali)-[~/Dos/assgn1]
$ nano faop.sh

(munu@kali)-[~/Dos/assgn1]
$ chmod +x faop.sh

(munu@kali)-[~/Dos/assgn1]
$ ./faop.sh
enter two numbers
5.6
2.4
Sum: 8.0
Difference: 3.2
Product: 13.44
Quotient: 2.33
Remainder: .008

```

```

GNU nano 8.1 faop.sh
echo "enter two numbers"
read num1
read num2

sum=$(echo "scale=2; $num1 + $num2" | bc)
echo "Sum: $sum"
diff=$(echo "scale=2; $num1 - $num2" | bc)
echo "Difference: $diff"
prod=$(echo "scale=2; $num1 * $num2" | bc)
echo "Product: $prod"
quo=$(echo "scale=2; $num1 / $num2" | bc)
echo "Quotient: $quo"
rem=$(echo "scale=2; $num1 % $num2" | bc)
echo "Remainder: $rem"

```

3. Ramesh's basic salary is input through the keyboard. His dearness allowance is 40% of basic salary, and house rent allowance is 20% of basic salary. Write a program to calculate his gross salary.

```

(munu@kali)-[~/Dos/assgn1]
$ nano salary.sh

(munu@kali)-[~/Dos/assgn1]
$ chmod +x salary.sh

(munu@kali)-[~/Dos/assgn1]
$ ./salary.sh
enter ramesh's basic salary
20000
Your dearness allowance is: 8000.00
Your house rent allowance is: 4000.00
Your gross salary is: 32000.00

```

```

GNU nano 8.1 salary.sh *
echo "enter ramesh's basic salary"
read bas_sal
DA=$(echo "scale=2; 0.40 * $bas_sal" | bc)
echo "Your dearness allowance is: $DA"
HRA=$(echo "scale=2; 0.20 * $bas_sal" | bc)
echo "Your house rent allowance is: $HRA"
gross=$(echo "scale=2; $bas_sal + $DA + $HRA" | bc)
echo "Your gross salary is: $gross"
5 HRA=$(echo "scale=2; 0.20 * $bas_sal" | bc)
6 echo "Your house rent allowance is: $HRA"

```

4. If a five digit number is input given through the keyboard during runtime, write a program to calculate the sum of its digits.

```

(munu@kali)-[~/Dos/assgn1]
$ chmod +x sumofdigits.sh

(munu@kali)-[~/Dos/assgn1]
$ nano sumofdigits.sh

(munu@kali)-[~/Dos/assgn1]
$ chmod +x sumofdigits.sh

(munu@kali)-[~/Dos/assgn1]
$ ./sumofdigits.sh
Enter a five digit number
12345
Sum : 15

```

```

GNU nano 8.1 sumofdigits.sh
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB

echo "Enter a five digit number"
read n
sum=0
while [ $n -gt 0 ]; do
r=$(( $n % 10 ))
sum=$(( $sum + $r ))
n=$(( $n / 10 ))
done
echo "Sum : $sum"

```

5. If cost price and selling price of an item is input through the keyboard, write a program to determine whether the seller has made profit or incurred loss. Also determine how much profit was made or loss incurred.

```

(munu@kali)-[~/Dos/assgn1]
$ nano profitloss.sh

(munu@kali)-[~/Dos/assgn1]
$ chmod +x profitloss.sh

(munu@kali)-[~/Dos/assgn1]
$ ./profitloss.sh
enter cost price
200
enter selling price
300
The seller has made a profit of : 100
profit percentage: 50.00 %

```

```

GNU nano 8.1 profitloss.sh
echo "enter cost price"
read CP
echo "enter selling price"
read SP
profit=$((SP-CP))
loss=$((CP-SP))
if [ $profit -gt 0 ]; then
echo "The seller has made a profit of : $profit"
profit_per=$((echo "scale=2; $profit * 100 / $CP" | bc ))
echo "profit percentage: $profit_per %"
elif [ $loss -gt 0 ]; then
echo "The seller has made a loss"
loss_per=$((echo "scale=2; $loss * 100 / $CP" | bc ))
echo "Loss percentage: $loss_per %"
else
echo "neither profit nor loss"
fi

```

6. Write a shell script which receives any year from the keyboard and determines, whether the year is a leap year or not. If no argument is supplied the current year should be assumed.

```

(munu@kali)-[~/Dos/assgn1]
$ nano leapyear.sh

(munu@kali)-[~/Dos/assgn1]
$ chmod +x leapyear.sh

(munu@kali)-[~/Dos/assgn1]
$ ./leapyear.sh
Enter a year :
2022
2022 is not a leap year .

```

```
GNU nano 8.1 leapyear.sh
#!/bin/bash
echo "Enter a year : "
read year
if (( ($year % 4 == 0 && $year % 100 != 0) || $year % 400 == 0 )); then
    echo "$year is a leap year . "
else
    echo "$year is not a leap year . "
fi
```

7. Write a shell script **allow** that will display a message to enter internal mark and percentage in attendance, if the entered mark is greater than equal to 20 and entered percentage in attendance is greater that equal to 75 then display the message Allowed for Semester otherwise display the message Not allowed.

```
(munu@kali)-[~/Dos/assgn1]
$ nano allow.sh

(munu@kali)-[~/Dos/assgn1]
$ chmod +x allow.sh

(munu@kali)-[~/Dos/assgn1]
$ ./allow.sh
Enter marks : 20 | && | $attendance -ge 75
23
Enter Attendance in Percentage :
75
Allowed
```

```
GNU nano 8.1 allow.sh
echo "Enter marks : "
read marks
echo "Enter Attendance in Percentage : "
read attendance
if [ $marks -ge 20 ] && [ $attendance -ge 75 ]; then
    echo "Allowed "
else
    echo "Not Allowed "
fi
```

8. Write a shell script **small3** that will compare three numbers passed as command line arguments and display the smallest one.

```

(munu@kali)-[~/Dos/assgn1]
$ nano small3.sh

(munu@kali)-[~/Dos/assgn1]
$ chmod +x small3.sh

(munu@kali)-[~/Dos/assgn1]
$ ./small3.sh
Enter Number :
5
Enter Number :
4
Enter Number :
5
Smallest : 4

```

```

GNU nano 8.1 small3.sh
echo "Enter Number : "
read num1
echo "Enter Number : "
read num2
echo "Enter Number : "
read num3
if [ $num1 -le $num2 ] && [ $num1 -le $num3 ]; then
    echo "Smallest : $num1 "
elif [ $num2 -le $num1 ] && [ $num2 -le $num3 ]; then
    echo "Smallest : $num2 "
else
    echo "Smallest : $num3 "
fi

```

9. Write a shell script **check_char** which will display one message to enter a character and according to the character entered it will display appropriate message from the following options:

- You entered a lower case alphabet
- You entered an upper case alphabet.
- You have entered a digit.
- You have entered a special symbol.
- You have entered more than one character.

```

(munu@kali)-[~/Dos/assgn1]
$ nano check_char.sh

(munu@kali)-[~/Dos/assgn1]
$ chmod +x check_char.sh

(munu@kali)-[~/Dos/assgn1]
$ ./check_char.sh
Enter A Character:
A
It is of Upper Case

```



```

GNU nano 8.1                                     check_char.sh
#!/bin/bash
echo "Enter A Character: "
read char
if [[ ${#char} -gt 1 ]]; then
    echo "You Entered more than one character"
elif [[ "$char" =~ [a-z] ]]; then
    echo "It is of Lower case"
elif [[ "$char" =~ [A-Z] ]]; then
    echo "It is of Upper Case"
elif [[ "$char" =~ [0-9] ]]; then
    echo "It is a Number"
else
    echo "Special Character"
fi

```

10. Write a shell script **class_time** which will display one message to enter a day and according to the day entered it will display the DOS class time along with the room information or the message "No class on day_name" or "Holiday" for Sunday.

```

(munu@kali)-[~/Dos/assgn1]
$ nano class_time.sh

(munu@kali)-[~/Dos/assgn1]
$ chmod +x class_time.sh

(munu@kali)-[~/Dos/assgn1]
$ ./class_time.sh
Enter A Day:
monday
DOS class at Room No. 109 at 11:30

```

```

GNU nano 8.1                                     class_time.sh
echo "Enter A Day: "
read day
case "$day" in
    "monday" | "MONDAY")
        echo "DOS class at Room No. 109 at 11:30"
        ;;
    "tuesday" | "TUESDAY")
        echo "DOS class at Room No. 109 at 10:00"
        ;;
    "wednesday" | "WEDNESDAY")
        echo "DOS class at Room No. 109 at 12:00"
        ;;
    "thursday" | "THURSDAY")
        echo "DOS class at Room No. 108 at 09:00"
        ;;
    "friday" | "FRIDAY")
        echo "DOS class at Room No. 110 at 09:30"
        ;;
    "saturday" | "SATURDAY")
        echo "DOS class at Room No. 110 at 03:00"
        ;;
    "sunday" | "SUNDAY")
        echo "Holiday"
        ;;
    *)
        echo "Invalid Day"
        ;;
esac

```

11. Write a shell script **filechk** that will take two file names as command line arguments, and check whether the content of two files are same or not . If contents of two files are same, then it will display the message: Files filename1 and filename2 have same content.
then delete the second file
and display the message: So filename2 is deleted.
Otherwise display the message: Files filename1 and filename2 have different content.

```
(munu@kali)-[~/Dos/assgn1]
$ nano filechk.sh

(munu@kali)-[~/Dos/assgn1]
$ chmod +x filechk.sh

(munu@kali)-[~/Dos/assgn1]
$ ./filechk.sh leapyear.sh check_char.sh
leapyear.sh check_char.sh differ: byte 25, line 2
Files leapyear.sh and check_char.sh have different contents
```

```
GNU nano 8.1 filechk.sh
#!/bin/bash
if [ $# -ne 2 ]; then
    echo "please provide exactly two filenames."
    exit 1
fi
if cmp "$1" "$2"; then
    echo "Files $1 and $2 have the same content "
    rm "$2"
    echo "$2 is deleted "
else
    echo "Files $1 and $2 have different contents "
```

12. Write a shell script **calculator** that will take three command line arguments, where the first argument will specify the first operand, second argument will specify the operator and the third argument will specify the second operand and display the output of the arithmetic operation specified in the following format: op1 operator op2 = result .

If the arguments will be passed in any other sequence, it will display the message:

“Invalid input “

Enter input in following format: op1 operator op2

The symbols to be used for different operators are as follows:

Addition:	+	Subtraction:	-
Multiplication:	x	Division:	/
Modulo:	%	Exponent:	^


```

(munu@kali)-[~/Dos/assgn1]
$ nano calculator.sh
30 echo "Supported operators are: +, -, x, /, %, ^"
31
(munu@kali)-[~/Dos/assgn1]
$ chmod +x calculator.sh
33
34 echo "$op1 $operator $op2 = $result"
(munu@kali)-[~/Dos/assgn1]
$ ./calculator.sh 2 + 4
2 + 4 = 6

```

```

GNU nano 8.1 calculator.sh *
#!/bin/bash
op1=$1
operator=$2
op2=$3
case "$operator" in
+)
result=$(echo "$op1 + $op2" | bc -l)
;;
-)
result=$(echo "$op1 - $op2" | bc -l)
;;
x)
result=$(echo "$op1 * $op2" | bc -l)
;;
/)
if [ "$op2" = "0" ]; then
echo "Error: Division by zero is not allowed"
exit 1
fi
result=$(echo "$op1 / $op2" | bc -l)
;;
%)
result=$(echo "$op1 % $op2" | bc)
;;
^)
result=$(echo "$op1 ^ $op2" | bc -l)
;;
*)
echo "Invalid input: Unsupported operator"
echo "Supported operators are: +, -, x, /, %, ^"
exit 1
;;
esac
echo "$op1 $operator $op2 = $result"

```