

Shell Script: Integer Addition with Input Validation

Step 1: Create a directory first using `mkdir sum` and change the directory to the `sum` directory.

```
santhosh@ccb228d15332571: ~/Demo/sum
santhosh@ccb228d15332571:~/Demo$ mkdir sum
santhosh@ccb228d15332571:~/Demo$ cd sum
```

Step 2: Touch and nano a file named `arithmetic.sh`

```
santhosh@ccb228d15332571:~/Demo/sum$ touch arithmetic.sh
santhosh@ccb228d15332571:~/Demo/sum$ nano arithmetic.sh
```

Step 3: Code Purpose

1. Prompts the user: Asks the user to enter two integer values.
2. Input Validation: Checks if the user provided input for both values. If either input is empty, an error message is displayed, and the script exits.
3. Calculates Sum:
Calculates the sum of the two input integers using the `bc` command for arbitrary precision arithmetic.
Calculates the sum using the `expr` command for basic integer arithmetic.
4. Displays Results: Prints the calculated sum using both `bc` and `expr`.

```
santhosh@ccb228d15332571:~/Demo/sum$ cat arithmetic.sh
#Shell program to add two integer values
#and check if any input is given or not

#!/usr/bin/bash
read -p "Input1 : " inp1
if [[ -z $inp1 ]]
then
    echo "Input 1 cannot be empty, please enter an integer."
    exit
fi

read -p "Input2 : " inp2
if [[ -z $inp2 ]]
then
    echo "Input 2 cannot be empty, please enter an integer."
    exit
fi

bc_val=`echo "$inp1+$inp2" | bc`
echo "BC Value : $bc_val"

expr_val=`expr $inp1 + $inp2`
echo "EXPR Value : $expr_val"

santhosh@ccb228d15332571:~/Demo/sum$
```

Step 4: Code Explanation

- `#!/usr/bin/bash` specifies that the script should be executed with the Bash interpreter.
- `read -p "Input1 : " inp1`: Prompts the user to enter the first integer and stores it in the `inp1` variable.

- `read -p "Input2 : " inp2`: Prompts the user to enter the second integer and stores it in the `inp2` variable.
- `if [[-z $inp1]]`: Checks if `inp1` is empty. If empty, displays an error message and exits with an error code (1).
- `if [[-z $inp2]]`: Checks if `inp2` is empty. If empty, displays an error message and exits with an error code (1).
- `bc_val=$(echo "$inp1 + $inp2" | bc)`: Uses the `bc` command to calculate the sum with arbitrary precision and stores the result in `bc_val`.
- `expr_val=$(expr $inp1 + $inp2)`: Uses the `expr` command to calculate the sum with basic integer arithmetic and stores the result in `expr_val`.
- `echo "BC Value : $bc_val"`: Displays the sum calculated using `bc`.
- `echo "EXPR Value : $expr_val"`: Displays the sum calculated using `expr`.

Step 5: Running the file using `./arithmetic.sh`

```
santhosh@ccb228d15332571:~/Demo/sum$ ./arithmetic.sh
-bash: ./arithmetic.sh: Permission denied
```

Step 6: Providing the execut permissions to the arithmetic shell file.

```
santhosh@ccb228d15332571:~/Demo/sum$ chmod +x arithmetic.sh
```

Step 7: Output of the shell script.

```
santhosh@ccb228d15332571:~/Demo/sum$ ./arithmetic.sh
Input1 : 12
Input2 : 14
BC Value : 26
EXPR Value : 26
```