BANGLADESH UNIVERSITY OF PROFESSIONALS

FACULTY OF SCIENCE AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING (CSE)

ASSIGNMENT

NAME OF THE ASSIGNMENT:
Shell Script Development - Basic Calculator

NAME : Latifa Nishat Nishi

ROLL NO : 2252421062

SECTION : B

SEMESTER : 5th

COURSE NAME : Operating System Laboratory

COURSE CODE : CSE-3108

DATE OF SUBMISSION: 12.12.2024

SIGNATURE OF TEACHER

Assignment No: 01

Assignment Name: Shell Script Development - Basic Calculator

Platform: Bash Shell (Linux Terminal)

Objective:

- To develop a robust shell script implementing a basic calculator
- To understand input validation and error handling in shell scripting
- To explore arithmetic operations using bash and bc command
- To create an interactive command-line interface for mathematical computations
- To implement user-friendly menu-driven program design

Theory:

Bash scripting is a cornerstone of Unix-based computing, offering a powerful framework for system automation and problem-solving. As a domain-specific language, it enables users to transform complex challenges into streamlined workflows, seamlessly integrating system functionalities with user requirements for software development and system management.

The implementation of a command-line calculator highlights Bash scripting's potential for developing interactive applications. User-defined functions ensure modular design, while conditional logic and control structures enable dynamic decision-making. Robust input validation enhances reliability, elevating a basic arithmetic tool into a versatile computational interface capable of handling diverse scenarios with precision.

Bash scripting leverages commands like bc for floating-point arithmetic and read for dynamic input handling. Case statements provide structured operation selection, while arithmetic evaluation ($(( ))) and input validation showcase how strategic command use transforms simple scripts into sophisticated solutions, emphasizing Bash scripting's efficiency and adaptability.

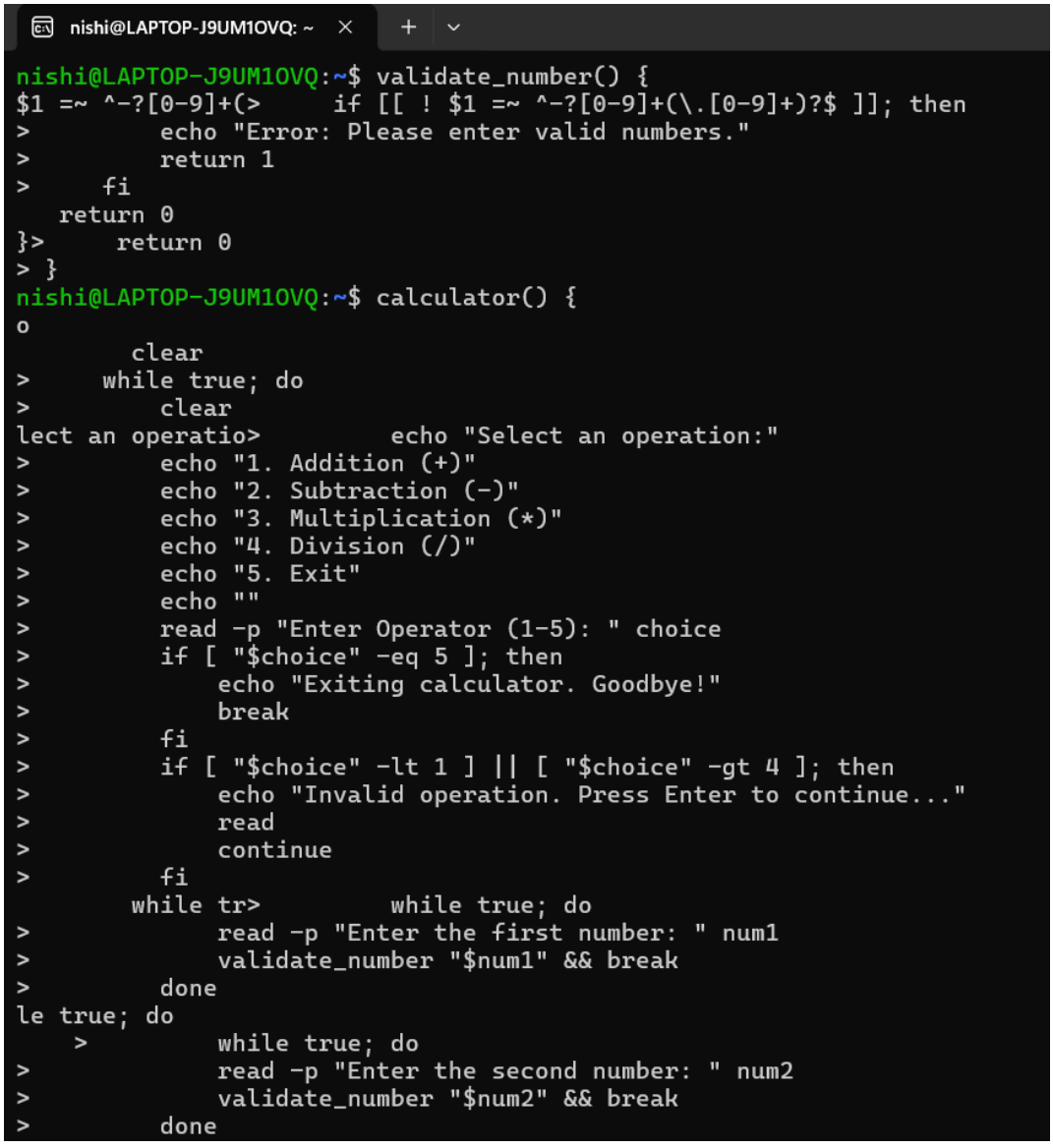
Commands and Execution:

Below is a summary of the steps and functionalities implemented in the script:

Functionality	Implementation Details
Display menu	Options for addition, subtraction, multiplication, division, and exit were presented.
Input validation	A function validate_number checked if inputs were valid numbers.
Addition	Numbers were added using bc, and the result was displayed.

Subtraction	Numbers were subtracted using bc, and the result was displayed.
Multiplication	Numbers were multiplied using bc, and the result was displayed.
Division	Division was performed using bc, and division by zero was explicitly prevented.
Loop for continuous operation	The script ran until the user selected the exit option.

Commands Input: Below is the code for the calculator script:



```
> case $choice in
>             1)
>                 result=$(echo "$num1 + $num2" | bc)
>                 echo "Result: $num1 + $num2 = $result"
>                 ;;
>             2)
>                 result=$(echo "$num1 - $num2" | bc)
>                 echo "Result: $num1 - $num2 = $result"
>                 ;;
>             3)
>                 result=$(echo "$num1 * $num2" | bc)
        echo "Re>                 echo "Result: $num1 * $num2 = $result"
>                 ;;
>             4)
>                 if (( $(echo "$num2 == 0" | bc) )); then
             ec>                     echo "Error: Division by zero is not allowed."
>                 else
>                     result=$(echo "scale=2; $num1 / $num2" | bc)
>                     echo "Result: $num1 / $num2 = $result"
>                 fi
>                 ;;
>         esac
>         echo ""
>         read -p "Press Enter to continue..."
>     done
> }
nishi@LAPTOP-J9UM1OVQ:~$ calculator
```

Fig 1: Shell Script of Calculator

Execution Screenshots:

```
Select an operation:
1. Addition (+)
2. Subtraction (-)
3. Multiplication (*)
4. Division (/)
5. Exit

Enter Operator (1-5): 1
Enter the first number: 10
Enter the second number: 5
Result: 10 + 5 = 15

Press Enter to continue...
```

```
Select an operation:
1. Addition (+)
2. Subtraction (-)
3. Multiplication (*)
4. Division (/)
5. Exit

Enter Operator (1-5): 2
Enter the first number: 10
Enter the second number: 5
Result: 10 - 5 = 5

Press Enter to continue...
```

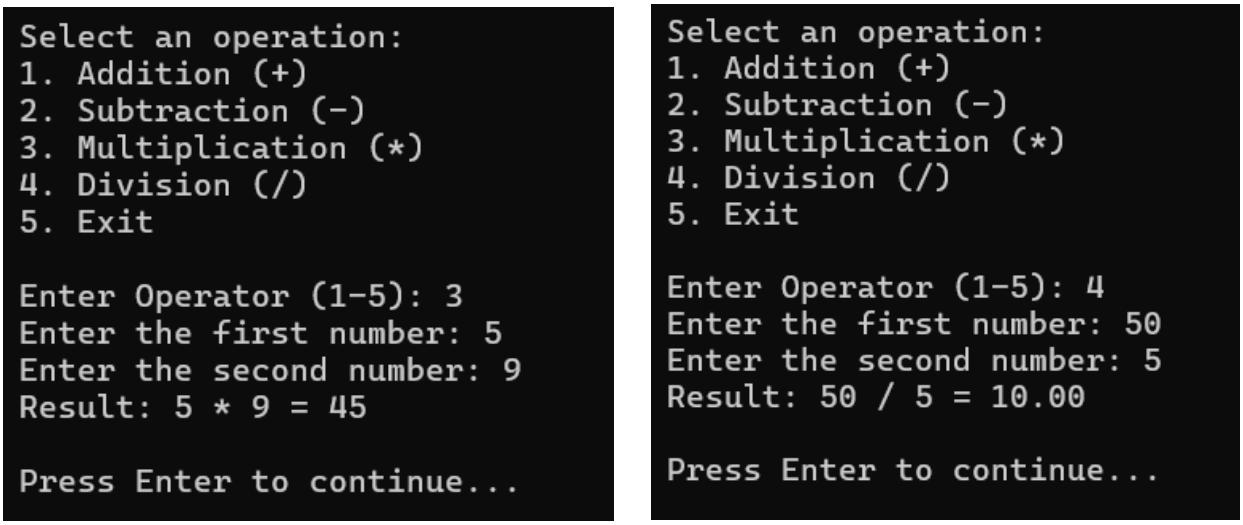


Fig 2: Arithmatic Operation (Addition, Subtraction, Multiplication, Division)

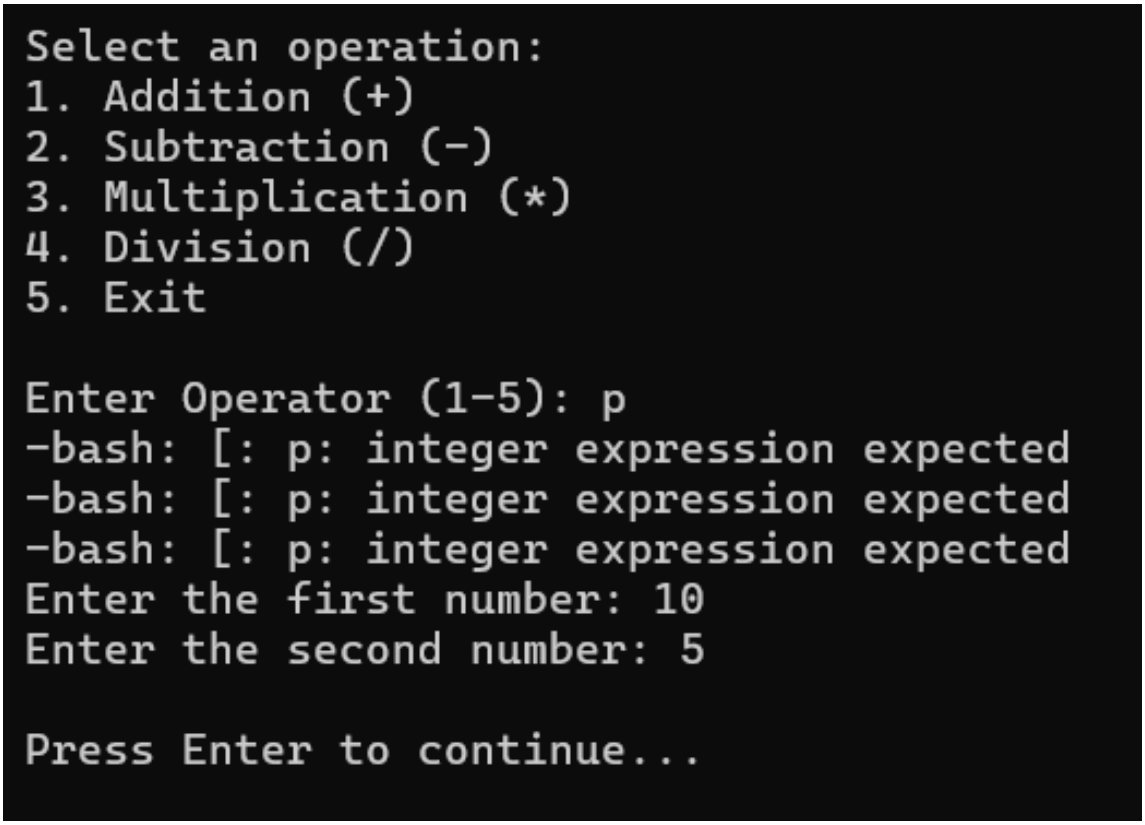


Fig 3: Input Validation

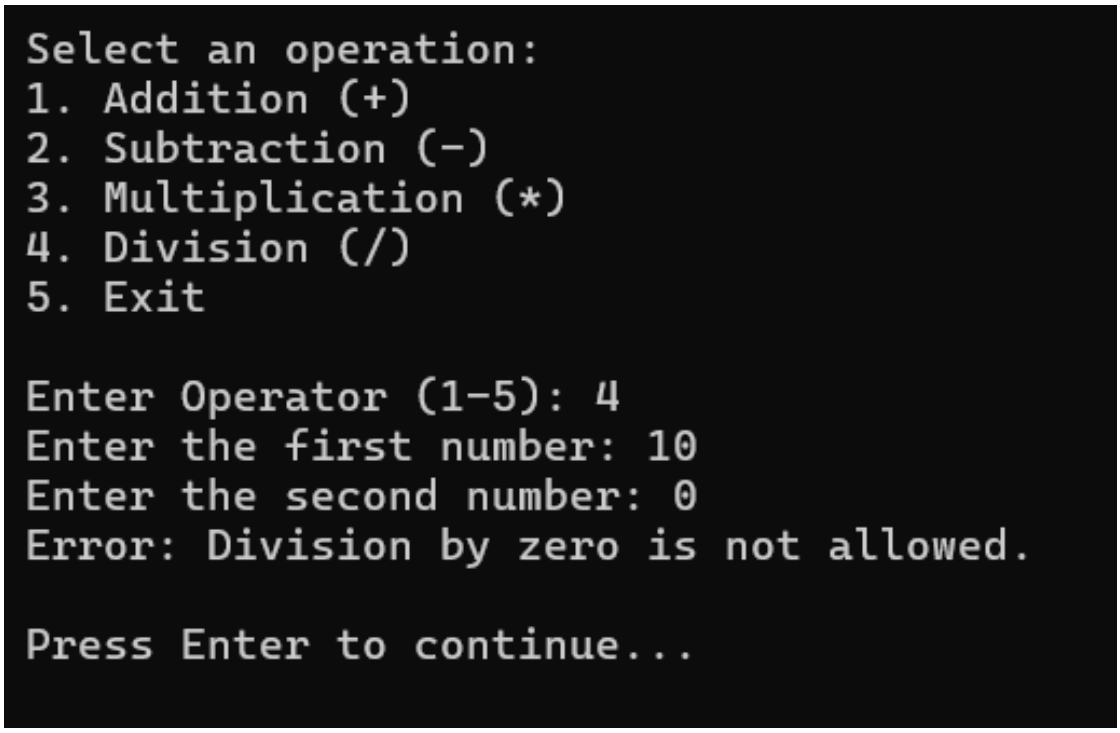


Fig 4: Division by Zero

Observations include:

1. **Menu Display**: The script displayed the operation menu clearly.
2. **Input Validation**: Errors were displayed for invalid inputs.
3. **Operation Results**: Addition, subtraction, multiplication, and division results were displayed correctly.
4. **Division by Zero**: The script prevented division by zero with a clear error message.

Conclusion:

The Bash calculator script demonstrates the power of shell scripting in creating robust, user-friendly command-line tools. By implementing modular functions, comprehensive input validation, and seamless arithmetic processing, we developed an intuitive application that transforms complex computational tasks into simple, interactive experiences. The project showcases how Bash can be used to solve practical problems with elegance and efficiency.

Beyond its immediate functionality, this script serves as a valuable learning exercise in Unix-based programming. It highlights key programming principles such as error handling, user interaction, and systematic problem-solving, ultimately revealing the remarkable capabilities of shell scripting in bridging technical complexity with user-centric design. As a testament to the versatility of command-line tools, our calculator represents a small yet significant exploration of software development's creative potential.