**Lab Practical Overview**

* **Duration:** 2 hours
* **Total Marks:** 30
* **Topics Covered:**
  1. Shell Scripting Fundamentals (variables, data types, operators, conditionals, loops).
  2. Advanced Shell Scripting (functions, parameter passing, file handling, error handling, regular expressions).

**Task A: Basic Shell Scripting Fundamentals (5 marks)**

1. **Create a Simple Script**
   * Write a shell script (hello.sh) that prints “Hello, World!” to the terminal.
   * Include a few echo statements to demonstrate printing messages.
2. **Demonstrate Variables and Operators**
   * Within the same script or a new one, declare at least two variables (e.g., num1 and num2) and perform basic arithmetic (addition, subtraction, multiplication, division).
   * Print the results.

**Evaluation (5 marks)**

* 2 marks: Correct creation of a functional shell script that runs without errors.
* 2 marks: Proper use of variables and arithmetic operators.
* 1 mark: Clarity and readability of the script (comments, structure).

**Task B: Conditional Statements and Loops (5 marks)**

1. **Conditional Logic**
   * Write a script (conditional.sh) that prompts the user for an integer input.
   * Use an if-else or case statement to check if the number is positive, negative, or zero, then print the result.
2. **Loops**
   * Extend the script or create a new one to demonstrate a loop (e.g., for, while) that iterates through a range of numbers and prints each number to the terminal.

**Evaluation (5 marks)**

* 2 marks: Proper use of conditional statements (if-else, case) with correct logic.
* 2 marks: Correct implementation of a loop (for or while) with clear output.
* 1 mark: Input validation or informative prompts/messages.

**Task C: Basic Automation Script (5 marks)**

1. **Task Automation**
   * Write a script (cleanup.sh) that automates a simple task, such as:
     + Finding and deleting temporary files (e.g., .tmp or .log files) in a given directory, **OR**
     + Archiving old files into a backup folder.
2. **Use of Command-Line Arguments**
   * Allow the user to specify the target directory or the file extension as a command-line argument (e.g., ./cleanup.sh /path/to/directory .log).

**Evaluation (5 marks)**

* 2 marks: Script runs successfully and performs the intended task (cleanup or backup).
* 2 marks: Proper handling of command-line arguments.
* 1 mark: Informative output indicating which files were affected.

**Task D: Advanced Shell Scripting Techniques (10 marks)**

1. **Functions and Parameter Passing**
   * Create a script (advanced.sh) containing at least **two functions**:
     + A function that accepts parameters (e.g., function greet\_user()) and prints a personalized greeting.
     + A function that performs a file or directory check (e.g., verifies if a given path exists).
2. **File Handling and I/O**
   * In the same script, demonstrate reading from a file line by line and writing output to a new file.
   * For example, read from input.txt and write to output.txt.
3. **Error Handling and Debugging**
   * Use set -x (or bash -x) for debugging to show command execution.
   * Implement a simple error-handling mechanism (e.g., checking if the file exists before reading it).
4. **Regular Expressions**
   * Include an example that uses grep or sed with a regular expression to search for a pattern in a file.
   * Print the matched lines or perform a text replacement.

**Evaluation (10 marks)**

* 2 marks: Correct implementation of functions with parameter passing.
* 2 marks: Proper file handling (reading and writing).
* 2 marks: Demonstration of error handling (checking file existence, debugging mode).
* 2 marks: Use of regular expressions (grep, sed, or similar).
* 2 marks: Overall script clarity, structure, and correctness of output.

**Task E: Reflection & Verification (5 marks)**

1. **Short Reflection**
   * In a separate text file (or at the end of advanced.sh as comments), describe:
     + Which shell features were most challenging and why.
     + How you debugged any errors.
     + How these scripts can be adapted to real-world use cases.
2. **Verification**
   * Show the instructor or evaluator that each script runs successfully:
     + hello.sh (or similar) for basics.
     + conditional.sh for conditionals/loops.
     + cleanup.sh for automation.
     + advanced.sh for advanced features.

**Evaluation (5 marks)**

* 3 marks: Clear and thoughtful reflection on the scripting process and challenges.
* 2 marks: Proper demonstration and verification of scripts running without errors.

**Suggested Timeline (2 Hours)**

1. **Task A & B (30 minutes)**
   * Basic scripting, variables, operators, conditionals, and loops.
2. **Task C (20 minutes)**
   * Basic automation script with command-line arguments.
3. **Task D (50 minutes)**
   * Advanced shell scripting: functions, file I/O, error handling, regex.
4. **Task E (20 minutes)**
   * Reflection, verification, and final submission.

**Final Notes**

* **Environment Setup**: Ensure you have a Linux environment (local machine, VM, or cloud instance) with a standard shell (e.g., Bash).
* **Submission**:
  + Provide all script files and any associated input files.
  + Include your reflection in a separate text file or at the bottom of your main script.
* **Grading Rubric**: Each task has a specific mark allocation; clarity, correctness, and completeness are key.