



SCHOOL OF ELECTRONICS ENGINEERING
CONTINUOUS ASSESSMENT TEST - I
WINTER SEMESTER 2025-2026

Programme Name & Branch : B.Tech VLSI Design
Course Code and Course Name : BEVD205L , **Scripting Languages and Verification**
Faculty Name(s) : Dr. J. Saikia
Class Number(s) : VL2025260500896
Date of Examination : 27/01/26
Exam Duration : 90 minutes **Maximum Marks:** 50

General instruction(s):

- Answer All Questions
- M - Max mark; CO – Course Outcome; BL – Blooms Taxonomy Level (1 – Remember, 2 – Understand, 3 – Apply, 4 – Analyse, 5 – Evaluate, 6 – Create)
- Course Outcomes:
 - CO1: Handle files, directories and manage processes using PERL scripts.
 - CO2: Handle files, directories and manage processes using TCL scripts.

Q. No	Question	M	CO	BL
1.	A) What will be the output of the following Perl code? Also, show how you arrived at the solution by going through each iteration. <pre> my @data = (2, 4, 6, 8); my \$sum = 0; foreach my \$x (@data) { if (\$x % 4 == 0) { \$sum += \$x; \$x = \$sum; } } print "@data"; </pre>	5	1	3
	B) Write a Perl script that: <ul style="list-style-type: none"> - Defines a subroutine max(\$a, \$b) that returns the larger of two numbers - Reads integers from standard input two at a time (one pair per line) - Calls the subroutine for each pair - Prints the two numbers followed by the maximum value <p>Example:</p> <p>Input: 3 7 10 4 5 5</p> <p>Output: 3 7 7 10 4 10 5 5 5</p>	5		
2.	Write a Perl script that: <ul style="list-style-type: none"> - Accepts one or more filenames from the command line - For each file, counts the number of lines matching the regular 	10	1	3



SCHOOL OF ELECTRONICS ENGINEERING
CONTINUOUS ASSESSMENT TEST - I
WINTER SEMESTER 2025-2026

	<p>expression /ERROR/ - Prints the filename and the corresponding error count Assume all files exist and are readable. Example: Command line: perl count_errors.pl a.log b.log a.log: ERROR failure\nERROR timeout Output: a.log 2</p>			
3.	<p>Explain how data handling and text processing are performed in Perl scripts. In your answer, discuss the role of the following Perl features and how they influence the way Perl scripts are written:</p> <ol style="list-style-type: none"> Scalars, arrays, and hashes Input handling mechanisms (@ARGV, STDIN, files) Regular expressions and pattern matching Line-oriented file processing (<>) Effect of data handling choices on script readability and correctness (You may use small examples where appropriate.) 	10	1	2
4.	<p>Write a Tcl script that should perform the following:</p> <ul style="list-style-type: none"> - Define a procedure <i>run_test</i> that takes: <ul style="list-style-type: none"> - a test name - an optional argument result with default value PASS - The procedure should print the test name and its result in the format: test_name : result Example: test2 : Pass - Outside the procedure in the main script <ul style="list-style-type: none"> - Read a test name from the user - Read a result from the user - If the user does not enter a result (empty input), call the procedure without the result argument - Otherwise, call the procedure with the user-provided result 	10	2	3
5.	<pre>set count 10 set student(name) "Ravi"</pre> <p>a) With reference to the above code, explain TCL arrays in detail. Describe how variables are created, modified, and removed.</p>	5	2	2
	<p>b) For the same example, describe how associative arrays are declared, accessed, and traversed. Support your answer with appropriate TCL syntax and examples.</p>	5		



**SCHOOL OF ELECTRONICS ENGINEERING
CONTINUOUS ASSESSMENT TEST - I
WINTER SEMESTER 2025-2026**

KEY

1)

A)

@data = (2, 4, 6, 8)

\$sum = 0

Iteration-wise:

Iteration#1: \$x = 2 → 2 % 4 != 0 → nothing happens

Iteration#2: \$x = 4 → 4 % 4 == 0

\$sum = 0 + 4 = 4

\$x = 4 → array unchanged

Iteration#3: \$x = 6 → 6 % 4 != 0 → nothing happens

Iteration#4: \$x = 8 → 8 % 4 == 0

\$sum = 4 + 8 = 12

\$x = 12 → modifies array element

Final array:

(2, 4, 6, 12)

B)

use strict;

use warnings;

```
sub max {
    my ($a, $b) = @_ ;
    return ($a > $b) ? $a : $b;
}
```

```
while (<STDIN>) {
    chomp;
    my ($x, $y) = split;
    my $m = max($x, $y);
    print "$x $y $m\n";
}
```

2)

use strict; use warnings;

```
foreach my $file (@ARGV) {
    open(my $fh, "<", $file) or die "Cannot open $file\n";
    my $count = 0;
    while (my $line = <$fh>) {
```



**SCHOOL OF ELECTRONICS ENGINEERING
CONTINUOUS ASSESSMENT TEST - I
WINTER SEMESTER 2025-2026**

```

if ($line =~ /ERROR/i) {
    $count++;
}
}
close($fh);
print "$file : $count\n";
}

```

4)

```

# Define the procedure
proc run_test {test_name {result PASS}} {
    puts "$test_name : $result"
}

# Read test name
puts "Specify Test case "
gets stdin test_name

# Read test result
puts "Test Result? "
gets stdin test_result

# Check if result is empty
if {$test_result eq ""} {
    # Call procedure with default result
    run_test $test_name
} else {
    # Call procedure with user-provided result
    run_test $test_name $test_result
}

```