Daffodil International University

Faculty of Science & Information Technology

Department of Computing and Information System

Final Examination, Spring-2026

Course Code: CIS115, Course Title: Structured Programming

Level: 1 Term: 1

Exam Duration: 2 Hours **Marks: 40**

Answer ALL Questions

[The figures in the right margin indicate the full marks and corresponding course outcomes. All portions of each question must be answered sequentially.]

1.	*a)*	**Explain** the different types of data types in C with examples.	[10]	CO1
	b)	**Outline** the defining characteristics of low-level, middle-level, and high-level programming languages.		
	c)	**Define** a C token and **state** all the fundamental types of tokens recognized by the compiler.		
	d)	**Explain** the difference between a keyword and an identifier with examples.		
	e)	**Compare** the advantages of using an algorithm over a flowchart.		
2.	*a)*	**Write** a C program utilizing a user-defined function to calculate and print the sum of all even digits of a positive integer entered by the user. For example, if the user inputs 4526, the program should calculate 4 + 2 + 6 and output 12.	[5]	CO4

	b)	**Develop** a C program that calculates the median of three integers entered by the user. Your program should produce output in the following format:
		SAMPLE OUTPUTS:
		`Enter three numbers: 45 27 39`
		`Median of 45, 27, and 39 is 39`
		`Enter three numbers: 67 145 44`
		`Median of 67, 145, and 44 is 67`
		`Enter three numbers: 15 25 35`
		`Median of 15, 25, and 35 is 25`
			[5]	

3.	*a)*	**Trace** the execution to determine how the variable **m** changes during each of the following independent C code fragments, and **calculate** its final integer value. i. `int j = 2, k = 6, m; m = ++j + k--;` ii. `int j = 2, k = 3, m; m = j + (1 > k - j);` iii. `int i = 1, j = 1, m; m = ++i && --j;` iv. `int j = 1, k = 6, m; m = --j ? --k : ++k;` v. `int i = 1, j = 3, m; m = (i < j) % j;`	[5]	CO2

	b)	**Trace** the following C programs and **compute** their exact final outputs.
		
		`#include <stdio.h>`
		
		`int main()`
		`{`
		`  int n = 10;`
		`  int j = 10;`
		
		`  for (n = 7; n >= 1; n = n - 2)`
		`  {`
		`  for (j = 0; j < n; j++)`
		`  printf(" %d", n - 1);`
		`  printf(" C\n");`
		`  }`
		
		`  return 0;`
		`}`
		
		`#include <stdio.h>`
		
		`int main() {`
		`  int x = 1, y = 2;`
		`  switch(x) {`
		`  case 1:`
		`  y++;`
		`  case 2:`
		`  if (y == 3) {`
		`  x += 2;`
		`  break;`
		`  }`
		`  case 3:`
		`  y += x;`
		`  }`
		`  printf("x=%d, y=%d", x, y);`
		`  return 0;`
		`}`
			[5]	

4.	*a)*	**Modify** the following program to correctly find the maximum value in an array by fixing all of its errors. In your answer, you must **identify** the types of errors, **write** the corrected C code, and **construct** a flowchart representing your corrected logic.
		
		`#include <stdio.h>`
		`int main() {`
		`  int arr(4) = {10, 50, 20, 40};`
		`  int max = 0;`
		`  for(int i = 1; i <= 4; i++) {`
		`  if(arr[i] > max) {`
		`  max = arr[i];`
		`  }`
		`  }`
		`  printf("The max is %f", max);`
		`  return 0;`
		`}`
			[6]	CO3

	b)	**Construct** a C program implementing a nested loop structure to produce the following number pattern.
		
		`1 2 3 4 5`
		`1 2 3 4`
		`1 2 3`
		`1 2`
		`1`
			[4]	