Daffodil International University
Department of Software Engineering
Faculty of Science & Information Technology
Final Examination, Spring 2025

Course Code:SE 121; Course Title: Structured Programming

Sections & Teachers: MTK(A, B), SA (C), SCS (D, E), AHZ (F, G) MSA (H), MK(I, J, M), JIC (K, L), MSSS (N), ST (O), MRN (P), NML (Q)

Time: 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) **Detect** any errors in the following codes and **Modify** them with the necessary changes. **[Marks-6]** **CLO-3** *Level-3*

i)

```
#include <stdio.h>
int main()
{
    int arr[3] = {1, 2, 3, 4};
    for (int i = 0; i < 4; i++)
        printf("%d ", arr[i]);
    return -1;
}
```

ii)

```
#include <stdio.h>
int main() {
    char q1[5], q2[5];
    gets(q1, q2);
    printf("You entered: %s
and %c", q1, q2);
    return;
}
```

b) Suppose you are working as a junior intern at a local weather station. Your task is to analyze the weekly temperature data to find the average temperature over 7 days. **[Marks-7]**

Construct a C program that uses an array to store daily temperatures (°C) over a week and compute the average temperature.

Sample Input:	Sample Output:
Enter temperature for Day 1: 30 Enter temperature for Day 2: 32 Enter temperature for Day 3: 31 Enter temperature for Day 4: 29 Enter temperature for Day 5: 35 Enter temperature for Day 6: 33 Enter temperature for Day 7: 28	Average Temperature = 31.14

	b)	Suppose you are building a simple registration system for a workshop. The system should take first and last names separately and then display the full name. **Construct** a C program that concatenates the first and last name and displays the full name. **Sample Input:** Enter First Name: Sarah, Enter Last Name: Khan **Sample Output:** Full Name: Sarah Khan	[Marks-7]	

2.	*a'*	**Identify** the output of the following C codes and **Explain** the reasoning behind your answer.
		
		i)
		`#include <stdio.h>`
		`int modify(int a)`
		`{`
		`  if (a % 3 == 0)`
		`  return a + 5;`
		`  else`
		`  return a - 2;`
		`}`
		`int main()`
		`{`
		`  int i, result = 0;`
		`  for(i = 1; i <= 5; i++)`
		`  {`
		`  result += modify(i);`
		`  }`
		`  printf("%d\n", result);`
		`  return 0;`
		`}`
		
		ii)
		`#include <stdio.h>`
		`int calculate(int x, int y)`
		`{`
		`  if (x > y)`
		`  return x - y;`
		`  else`
		`  return x + y;`
		`}`
		`int main()`
		`{`
		`  int p = 8, q = 5;`
		`  printf("Result = %d\n", calculate(p, q));`
		`  printf("Result = %d\n", calculate(q, p));`
		`  return 0;`
		`}`
			[Marks-6]	CLO-4
				Level-4

	b)	Suppose you are developing a math helper tool for primary school students. The students are having trouble recognizing the prime numbers. **Analyze** and **simulate** a function that checks whether the entered number is prime. **Sample Input:** Enter a number: 17 **Sample Output:** 17 is a prime number.	[Marks-7]	
	c)	Suppose you are simulating an ATM system that must validate withdrawal requests. To ensure proper denomination availability, the ATM only allows withdrawals that are divisible by 500 or 1000, and only if the amount is less than or equal to the account balance. **Analyze** and **simulate** the function ***int withdraw (int balance, int amount)*** to determine whether a withdrawal amount is valid based on ATM rules (divisibility by 500 or 1000 and sufficient balance). Update the balance accordingly and write the complete C program to implement this logic. **Sample Input:** Enter account balance: 5000, Enter withdrawal amount: 1000 **Sample Output:** Transaction Approved. Remaining Balance = 4000	[Marks-7]	