Daffodil International University

Faculty of Science & Information Technology

Department of Software Engineering

Final Examination - Summer 2025

Course Code: SE 121; Course Title: Structured Programming

Sections & Teachers: A – M (MAK, ZNM, MR, JIC, SCS, AHZ, KFH, AF)

Time: 2:00 Hrs **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) **Examine** 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 (){
    float height[100];
    for(i=0; i<100; i++)
        scanf("%f", height[i]);

    //printing first element
    printf("%f", height[1]);
    //printing last element
    printf("%f", height [100]);

    return 0;
}
```

ii)
```
#include <stdio.h>
int main() {
    char username[8];
    char correctUser[] = "admin";

    printf("Enter username: ");
    scanf("%c", username);

    if (username == correctUser)
        printf("Welcome, admin!\n");

    return 0;
}
```

b) You are helping organize a running competition. Your task is to find the fastest completion time among all participants and the position of the participant who achieved it. **[Marks-7]**

If more than one participant has the same fastest time, display the position of the first one.

Construct a C program that takes the number of participants and their completion times as input, and outputs the fastest time along with the position (*1-indexed*) of the participant.

Sample Input:	**Sample Output:**
Enter number of participants: 5 Enter time for Participant 1: 20 Enter time for Participant 2: 25 Enter time for Participant 3: 18 Enter time for Participant 4: 22 Enter time for Participant 5: 18	Fastest Time = 18 Position = 3

c) Suppose you are a junior developer in a new social media platform. The platform wants to automatically create usernames for users by combining their first and last names without spaces and making them lowercase. As a developer of the system, you have to write the code. **[Marks-7]**

Now, **construct** a C program that takes a first name and last name from the user and outputs a lowercase username.

Sample Input:	**Sample Output:**
First name: Nazmus Last name: Sakib	Your username: nazmussakib

2. *a)* **Identify** the output of the following C codes and **explain** the reasoning behind your answer. **[Marks-6]** **CLO-4** ***Level-4***

i.
```
#include <stdio.h>
int tweak(int a) {
   if (a % 4 == 1 && a%5 != 0)
      return a * 2;
   else if (a % 2 == 0)
      return a /= 2;
   else return a - 1;
}
int main() {
   int i, result = 0;
   for (i = 2; i <= 7; i++) {
      result += tweak(i);
   }
   printf("%d\n", result);
   return 0;
}
```

ii.
```
#include <stdio.h>

int combine(int x, int y) {
   if (x == y)
      return x * y;
   else if ((x + y) % 2 == 0)
      return x + y;
   else
      return x – y;
}
int main() {
   int a = 6, b = 6, c = 3;
   printf("R1 = %d\n", combine(a, b));
   printf("R2 = %d\n", combine(a, c));
   printf("R3 = %d\n", combine(c, b));
   return 0;
}
```

b) Suppose you are developing a personalized learning assistant for a rural school where many students cannot afford printed multiplication charts. During math class, the teacher wants a quick way to show multiplication tables for any number so students can practice together. **[Marks-7]**

Now, **analyze** and **simulate** a *function* that generates and displays the multiplication table for the entered number.

Sample Input:	**Sample Output:**
Enter a number: 5	5 × 1 = 5 5 × 2 = 10 5 × 3 = 15 ... 5 × 10 = 50

c) Suppose you are simulating a food delivery startup's order payment system. The system must validate whether a customer's payment is acceptable. The startup only allows orders to be placed if the customer has sufficient balance in his/her wallet and the total bill amount is at least 200 Tk. If the payment is valid, deduct the bill from the customer's wallet balance. **[Marks-7]**

Analyze the scenario & write the complete C program to implement this logic using a *function*.

Sample Input:	**Sample Output:**
Enter wallet balance: 1500 Enter bill amount: 500	Payment Successful. Remaining Wallet Balance = 1000

Sample Input:	**Sample Output:**
Enter wallet balance: 1500 Enter bill amount: 2000	Insufficient Balance. Remaining Wallet Balance = 1500

Sample Input:	**Sample Output:**
Enter wallet balance: 1500 Enter bill amount: 100	Minimum order amount is 200 Tk. Remaining Wallet Balance = 1500