



Institute of Geographical Information Systems

CS-212 - Object Oriented Programming LAB

Semester: Fall 2025

Class: SCEE-IGIS - 2024

Name: Ali Nawaz

CMS ID : 00000526123

Submitted to: Ma'am Alvina Anjum

Due Date: Sep 10, 2025

Lab 01: Functions

Question No 1:

Write a program using functions in which it asks the user to enter their data based on which the program decides whether the student got admission in NUST or not. The program takes name, CNIC, date of birth, NUST test % and FSc % as input. The criteria for admission is given below:

A student is given admission if the age is more than 16 years old and less than 25 and the aggregate percentage is greater than 75%. Aggregate percentage is calculated by taking 75% of FSc percentage and 25% of NUST entry test %. Assume that this program is written in year 2022.

If the student passes the criteria for admission, he/she is awarded scholarship based on following criteria

NUST Entry Test %	Scholarship
< 80	Nil
≥ 80	50%
≥ 90	100%

A sample output is shown below:

Name: Hassan

CNIC: 37405-2225477-0

DOB: 20-10-2005

Test: 80 %

FSc: 80%

Admission: Yes

Scholarship: 50%

Screenshot:

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 int calculateAge(int birthYear, int currentYear)
6 {
7     return currentYear - birthYear;
8 }
9
10 float calculateAggregate(float fsc, float test)
11 {
12     return (0.75 * fsc) + (0.25 * test);
13 }
14
15 string scholarship(float testPercent)
16 {
17     if (testPercent >= 90)
18         return "100%";
19     else if (testPercent >= 80)
20         return "50%";
21     else
22         return "Nil";
23 }
24
25 int main()
26 {
27     string name, cnic, dob;    "cnic": Unknown word.
28     float fsc, test;
29     int birthDay, birthMonth, birthYear;
30     char dash;
31 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER GITLENS

```
● alinawaz@Ali-MacBook-Air: Week-01 % cd "/Users/alinawaz/Developer/Development/OOP/Week-01/" && g++ Problem1.cpp -o Problem1 && "/Users/alinawaz/Developer/Development/OOP/Week-01/"Problem1
Enter Name: Ali Nawaz
Enter CNIC: 34101-1234567-8
Enter DOB: 1-2-2003
Enter Test %: 90-99
Enter NUST Test %: 80-89
Enter FSC %: 85 %
Admission: Yes
Scholarship: 50%
● alinawaz@Ali-MacBook-Air: Week-01 %
```

In 68, Col 2 Spaces: 4 UTF-8 LF ⓘ C++ ⌂ Go Live Mac ⓘ Prettier ⌂

Question No 2:

In the Counting Poetry Slam, you start at 1 and want to reach a number N.

- Normally, you just add 1 each time.
 - But you have a twist: at most once, you can reverse the digits of the current number.
- Write a program that calculates the minimum steps needed to reach N using these rules.

Input:

- A single integer N (the target number).

Output:

- The minimum steps required.

Example 1:

Input: 19

Output: 19

Explanation: Just count from 1 to 19.

Example 2:

Input: 23

Output: 13

Explanation: 1 → 2 → ... → 12 → reverse(12=21) → 22 → 23

Screenshot:

The screenshot shows a Visual Studio Code interface with the following details:

- Code Editor:** The main area displays a C++ file named Problem2.cpp. The code defines a function `reverseNumber` that takes an integer `n` and returns its reverse. It also contains a `main` function that prompts the user for a number `N`, initializes variables `steps`, `current`, and `usedReverse`, and then enters a loop where it repeatedly calls `reverseNumber` on `current` until `current` is greater than or equal to `N`. The code uses standard input/output streams (`<iostream>`) and the `std` namespace.
- Terminal:** At the bottom, the terminal window shows the command `g++ Problem2.cpp -o Problem2 && ./Problem2` being run in a Mac OS X terminal.
- Explorer Sidebar:** On the right, the sidebar shows the project structure for "Week-01". It includes files like `Problem1.cpp`, `Problem2.cpp`, `Problem3.cpp`, `CMakeLists.txt`, and `main.cpp`. It also shows a PDF file `OOP Lab01 - 526123.pdf` and a folder `cmake-build-debug`.

Question No 3:

When you invest money in a savings account your money earns a certain amount of interest over a period of time. For example, investing \$100 in an account that yields 1% interest annually would mean that after 1 year your account would be worth \$101.

If you left this money in your account you would earn additional interest in the following year due to the fact that your account now has \$101 instead of the initial deposit of \$100. You would earn interest on the initial deposit plus the \$1 you earned in interest—this is called “compound interest.”

For this problem, you will write a program that calculates how much a person can earn by investing in a high-yield savings account. The program should ask the user for an initial deposit amount and an interest rate. The program will then generate a 3-month projection that illustrates how much money the user can expect to earn. Assume that interest is compounded monthly (i.e. if your account earns a rate of 12% interest annually then you would earn a rate of 1% per month). Here's a sample running of the program:

```
This program will project how much you can earn by investing money in a high-yield savings account over a 3-month period.  
To begin, enter how much money you would like to initially invest (i.e. 500): 500  
Next, enter your projected annual interest rate. For example, enter 0.05 for 5%: 0.05  
  
Calculating . . .  
  
Month Starting Balance    Interest    Ending Balance  
1      500.00              2.08        502.08  
2      502.08              2.09        504.18  
3      504.18              2.10        506.28
```

Screenshot:

The screenshot shows a Microsoft Visual Studio Code interface. The main area displays a C++ code editor with the following code:

```
1 #include <iostream>
2 using namespace std;
3
4 int reverse_digits(int n)
5 {
6     int reversed_n = 0;
7     while (n > 0)
8     {
9         int digit = n % 10;
10        reversed_n = reversed_n * 10 + digit;
11        n /= 10;
12    }
13    return reversed_n;
14 }
15
16 int main()
17 {
18     int N;
19     cin >> N;
20
21     int min_steps = N - 1;
22
23     for (int i = 1; i <= N; ++i)
24     {
25         int reversed_i = reverse_digits(i);
26
27         if (reversed_i <= N)
28     {
```

The code implements a function to reverse digits of an integer and uses it in a loop to find the minimum number of steps required to reach a given number N from 1.

The terminal tab shows the following command-line session:

```
alinawaz@Alis-MacBook-Air Week-01 % cd "/Users/alinawaz/Developer/Development/OOP/Week-01/" && g++ Problem2.cpp -o Problem2 && "/Users/alinawaz/Developer/Development/OOP/Week-01/"Problem2
23
13
alinawaz@Alis-MacBook-Air Week-01 % cd "/Users/alinawaz/Developer/Development/OOP/Week-01/" && g++ Problem2.cpp -o Problem2 && "/Users/alinawaz/Developer/Development/OOP/Week-01/"Problem2
21
11
alinawaz@Alis-MacBook-Air Week-01 %
```

The file explorer sidebar shows the project structure:

- OPEN EDITORS
- DEVELOPMENT
 - CMake
 - Jargon
 - NUSTVerse
 - OOP
 - Idea
 - cmake-build-debug
 - Week-01
 - OOP Lab01 - 526123.pdf
 - Problem1
 - Problem2
 - Problem3
 - Problem3.cpp
 - tempCodeRunnerFile.cpp
 - Week-02
 - CMakeLists.txt
 - main.cpp
 - Week01-526123.zip