

Lab 1

CS 313: Intermediate Computer Programming

Date: September 19, 2024

Lab Objectives and Instructions:

The goal of this lab is to touch on the concepts we've seen in class. Each section is designed to test your understanding ensuring that you grasp the fundamentals of C++ programming.

You are to attempt each question individually.

Section A: Direct Questions

1. Identify error(s), if any, in the following array declarations. If a statement is incorrect, provide the correct statement.

- a. `int primeNum[99];`
- b. `int testScores[0];`
- c. `string names[60];`
- d. `int list100[0..99];`
- e. `double[50] gpa;`
- f. `const double LENGTH = 26;`
`double list[LENGTH - 1];`
- g. `const long SIZE = 100;`
`int list[2 * SIZE];`

2. Determine whether the following array declarations are valid. If a declaration is invalid, explain why.

- a. `int list[61];`
- b. `strings names[20];`
- c. `double gpa[];`
- d. `double[-50] ratings[];`
- e. `string flowers[35];`
- f. `int SIZE = 10;`
`double sales[2 * SIZE];`
- g. `int MAX_SIZE = 50;`
`double sales[100 - 2 * MAX_SIZE];`

3. Without using computer, what is the output of the following program segment?

```
double list[5];

for (int i = 0; i < 5; i++)
    list[i] = pow(i, 3) + i / 2.0;

cout << fixed << showpoint << setprecision(2);
for (int i = 0; i < 5; i++)
    cout << list[i] << " ";
cout << endl;

list[0] = list[4] - list[2];
```

```
list[2] = list[3] + list[1];

for (int i = 0; i < 5; i++)
    cout << list[i] << " ";

cout << endl;
```

4. Determine whether the following array declarations are valid. If a declaration is valid, determine the size of the array.

- a. `int list[] = {18,13,14,16};`
- b. `int x[10] = {1,7,5,3,2,8};`
- c. `double y[4] = {2.0,5.0,8.0,11.0,14.0};`
- d. `double lengths[] = {8.2,3.9,6.4,5.7,7.3};`
- e. `int list[7] = {12,13,,14,16,,8};`
- f. `string name[8] = {"John", "Lisa", "Chris", "Katie"};`

Table 1: Reference Table for Programming Challenge Question 1

Group Score	Mid Point (x)	Frequency (f)	fx	$(x - \bar{x})^2$	$f(x - \bar{x})^2$
0-24	12	5			
25-49	37	8			
.	.	.			
.	.	.			
		$\sum f$	$\sum fx$		$\sum f(x - \bar{x})^2$

Candidate Name	A	B	C	D	E	Total Votes	% of Votes
Jamal-Deen Abdulai							
Total Votes							

Table 2: Reference Output table for Programming Challenge Question 2

Section B: Programming Challenge [60 pts]

- Write a program that reads students' test scores in the range 0–200 as user input. It should then determine the number of students having scores in each of the following ranges: **0–24, 25–49, 50–74, 75–99, 100–124, 125–149, 150–174, and 175–200**. Output the score ranges and the number of students. (Run your program with the following input data: 76, 89, 150, 135, 200, 76, 12, 100, 150, 28, 178, 189, 167, 200, 175, 150, 87, 99, 129, 149, 176, 200, 87, 35, 157, 189.)

Your program should output the mean, variance and standard deviation. Make use of appropriate functions in your program. Your output should be formatted as shown in **Table 1**

Useful formula:

$$\text{Mean} = \bar{x} = \frac{\sum fx}{\sum f} \quad \text{Variance} = \sigma^2 = \frac{\sum f(x - \bar{x})^2}{(\sum f) - 1} \quad \text{Standard Deviation} = \sigma = \sqrt{\frac{\sum f(x - \bar{x})^2}{(\sum f) - 1}}$$

- Write a program that allows the user to enter the full names of five candidates in a local election and the number of votes received by each candidate at each of the five polling stations (A, B, C, ... E). The program should then output each candidate's name, the number of votes received, and the percentage of the total votes received by the candidate. Your program should also output the results of the election in a sorted form starting with the candidate with the highest received votes. Make use of appropriate functions in your program.

A sample output is shown in **Table 2**

- Write a program that reads students' names followed by their test scores. The program should output each student's name followed by their test scores and the relevant grade. It should also find and print the highest test score and the name of the student(s) having the highest test score.

Student data should be stored in a `struct` variable of type **studentType**, which has four components; **studentFName** and **studentLName** of type `string`, **testScore** of type `int` (testScore is between 0 and 100), and **grade** of type `char`. Suppose that the class has 20 students, use an array of 20 components of type **studentType**.

Your program must contain the following function:

- A function to read the students' data into the array.
- A function to assign the relevant grade to each student.
- A function to find the highest test score.
- A function to print the name of the students having the highest test score.