**TRINITY INTERNATIONAL SS & COLLEGE**

**Dillibazar Height, Kathmandu, Nepal**

****

**LAB WORK #2: C-Programming (Structure &Union)**

**(COMPUTER SCIENCE)**

**SUBMITTED BY: SUBMITTED TO:**

**NAME: Prashim Timsina**

**GRADE: XII (MA2)**

**DATE : [2023/08/11] PRAVEEN KOIRALA**

**Faculty of Computer Science**

**KATHMANDU, NEPAL**

**2023 Table of Contents**

S. No. Page No.

1. Table of Contents 1

2. Objectives 2

3. Theoretical Background 2

3.1 C-Programming

3.1.1 Structure Concept

3.1.2 Union Concept

4. Work Done

5. Conclusion

Page: 1

**2. Objective**

The main objectives of the lab work are as follows:

1. To become familiar with structures.
2. To become familiar with union.
3. To understand the concept of typedef structures.
4. To understand and familiarize with the concept of using typedef to declare new names for existing variable in C-program.
5. To understand concept of nested structure

**3. Theoretical Background**

Introduction:

C programming offers structures and unions as versatile tools for managing data efficiently. This theory provides a brief exploration of their definitions, syntax, and applications.

Structures:

Structures are composite data types that group variables of diverse types under a single name, aiding in organized data representation.

Access structure members using the dot operator (.).

Syntax of Structures:

struct structure\_name {

data\_type member1;

data\_type member2;

// ... more members

};

Page: 2

Unions:

Unions, like structures, group variables together, but all members share the same memory space, suitable for exclusive data storage.

Access one union member at a time due to shared memory.

Syntax of Union:

union union\_name {

data\_type member1;

data\_type member2;

// ... more members

};

Q1. WAP to input id, name and grade for 10 students. Then print them. Use array of structure concept.

|  |
| --- |
| Source Code |

Q2. WAP to input 20 employees name, position and salary. Then search a record of an employee and its position on the basis of name.

|  |
| --- |
| Source Code |

Q3. WAP to input employees’ details with salary then print them. Use nested struct concept.

|  |
| --- |
| Source Code |

Q4. WAP to input id, name and address of 20 students using struct. Then print them in sorted format on the basis of name.

|  |
| --- |
| Source Code |

Q5. WAP to input any 10 teacher’s id, name and subject using ‘typedef’ and structure. Then print them on screen.

|  |
| --- |
| Source Code |

Q6. WAP using function and structure to calculate sum of two distances measured in terms of kilometers and meters.

|  |
| --- |
| Source Code |

Q7. WAP to input student id, name and grade and print them. Use union concept.

|  |
| --- |
| Source Code |

**5. Conclusion**

In the culmination of our lab work exploring structures and unions, coupled with the insightful application of typedef and nested structures, we've gained a profound appreciation for the art of data organization and memory management in C programming. Through typedef, we've streamlined our code and enhanced readability, enabling us to craft intuitive aliases for complex data types. Additionally, the utilization of nested structures has allowed us to model intricate relationships between data components, fostering a deeper understanding of real-world entities within our code. As we conclude this lab journey, we emerge with not only a mastery of these fundamental constructs but also a solid foundation for crafting efficient, elegant, and logically structured programs in C.