

4 Lecture - CS410

Important Subjective

1. Question: What is a structure in C/C++?

Answer: A structure in C/C++ is a user-defined data type that allows grouping multiple variables of different data types under a single name.

2. Question: How do you declare a structure in C/C++?

Answer: To declare a structure, you use the `struct` keyword followed by the structure's name and a list of its member variables.

3. Question: What is the difference between a structure and an array in C/C++?

Answer: Unlike an array, a structure can hold variables of different data types, while all elements in an array must be of the same data type.

4. Question: How do you access a member inside a structure in C/C++?

Answer: You can access a member inside a structure using the dot (.) operator followed by the member's name.

5. Question: What is a union in C/C++?

Answer: A union is a user-defined data type that allows multiple variables to share the same memory space, helping to save memory when only one variable is used at a time.

6. Question: What happens if you modify one member of a union and then access another member?

Answer: Modifying one member and accessing another member of a union results in undefined behavior. The value retrieved will depend on the memory layout and can lead to unexpected results.

7. Question: How do you declare a union in C/C++?

Answer: To declare a union, you use the `union` keyword followed by the union's name and a list of its member variables.

8. Question: Can a union have functions as its members in C/C++?

Answer: No, unions can only have variables as their members, not functions.

9. Question: Can a structure have pointers as its members in C/C++?

Answer: Yes, a structure can have pointers as its members, allowing it to store memory addresses of other variables.

10. Question: What is the primary difference between a structure and a union in C/C++?

Answer: The primary difference is that a structure allocates memory for each of its members separately, while a union shares the same memory space for all its members, allowing only one member to be active at any given time.