■ Data Types in C - Notes

In C programming, data types specify the type of data that a variable can store. C supports a rich set of data types that allow the programmer to select the appropriate type for the data being used.

■ Basic Data Types: 1. int - stores integers (whole numbers), without decimals. 2. float - stores floating-point numbers (numbers with decimals). 3. char - stores single characters. 4. double - stores large floating-point numbers with double precision.

■ Integer Types: - short int - int - long int - long long int These can be either signed (default) or unsigned. Example: unsigned int age = 25;

■ Floating Point Types: - float: 4 bytes, 6-7 decimal digits precision - double: 8 bytes, 15 decimal digits precision - long double: more precision and size (compiler dependent)

■ Character Type: - char: stores a single character using 1 byte. Example: char letter = 'A';

■ Derived Data Types: - Array - Pointer - Structure - Union

■ Enumeration (enum): - A user-defined type consisting of named integer constants. Example: enum Days { SUNDAY, MONDAY, TUESDAY };

■ Void Type: - void: represents no value or no type. - Used for functions that do not return a value. Example: void display();

■ Type Modifiers: - signed, unsigned, long, short - Modify the size and range of basic data types. Example: long int population; unsigned char level;

■ Example Program: #include <stdio.h> int main() { int a = 10; float b = 5.75; char c = 'Z'; double d = 123456.789; printf("Integer: %d\n", a); printf("Float: %.2f\n", b); printf("Character: %c\n", c); printf("Double: %.3lf\n", d); return 0; }

■ Output: Integer: 10 Float: 5.75 Character: Z Double: 123456.789

■ Summary: Choosing the right data type is crucial for efficient memory usage and program performance. Always consider the range and type of values you need to store.