Lecture – 8 –

Decrement Operator

- The decrement operator (--) works in much the same way as the increment operator, but instead of increasing the value, it decreases it by one.

```
--x; // prefix
x--; // postfix
```

Data Types

- The operating system allocates memory and selects what will be stored in the reserved memory based on the variable's data type.
- The data type defines the proper use of an identifier, what kind of data can be stored, and which types of operations can be performed.
- There are several built-in types in C++.

Expressions

- The examples below show legal and illegal C++ expressions.
- 55+15 // **legal** C++ expression (Both operands of the + operator are integers)
- 55 + "John" // **illegal** (The + operator is not defined for integer and string)
- "Hello," + "John" // **legal** (The + operator is used for string concatenation)

Numeric Data Types

- Numeric data types include:
  - Integers (whole numbers), such as -7, 42.
  - Floating point numbers, such as 3.14, -42.67.
  - Strings & Characters
  - Booleans

Integer

- **The integer** type holds non-fractional numbers, which can be positive or negative.
- **Examples:** 42, -42, and similar numbers.
- The size of the integer type varies according to the architecture of the system on which the program runs, although 4 bytes is the minimum size in most modern system architectures.
- Use the int keyword to define the integer data type. **int a = 42;**
- Several of the basic types, including integers, can be modified using one or more of these type modifiers:
    - **signed:** can hold both negative and positive numbers.
    - **unsigned:** can hold only positive values.
    - **short:** Half of the default size.
    - **long:** Twice the default size.
- Example: **unsigned long int a;**

Floating Point Numbers

- A floating-point type variable can hold a real number, such as 420.0, -3.33, or 0.03325.
- There are <u>three</u> different floating-point data types: float, double, and long double.
- In most modern architectures, a float is 4 bytes, a double is 8, and a long double can be equivalent to a double (8 bytes), or 16 bytes.
- For example: double temp = 4.21;
- Floating point data types are always signed, which means that they have the capability to hold both positive and negative values.

Strings

- **A string** is composed of numbers, characters, or symbols.
- String literals are placed in double quotation marks.
- **Examples:** "Hello", "My name is David".
- You need to include the <string> library to use the string data type.
- You don't need to include <string> separately, if you already use <iostream>.

```
#include <string>
using namespace std;
int main() {
string a = "I am learning C++";
return 0;
}
```

Characters

- **Characters** are single letters or symbols, and must be enclosed between single quotes, like 'a', 'b', etc.
- In C++, single quotation marks indicate a character;
- double quotes create a string literal.
- While 'a' is a single a character literal, "a" is a string literal.
- A char variable holds a 1-byte integer.
- However, instead of interpreting the value of the char as an integer, the value of a char variable is typically interpreted as an ASCII character.
- **Example:** char test = 'S';
- American Standard Code for Information Interchange (ASCII) is a character-encoding scheme that is used to represent text in computers.

Booleans

- **Boolean** data type returns just two possible values: true (1) and false (0).
- To declare a boolean variable, we use the keyword **bool**.
- bool online = false;
- bool logged_in = true;
- Conditional expressions are an example of Boolean data type.

Variable Naming Rules

- Use the following rules when naming variables:
    - All variable names must begin with a letter of the alphabet or an underscore(_).
    - After the initial letter, variable names can contain additional letters, as well as numbers.
    - Blank spaces or special characters are not allowed in variable names.
- C++ keyword (reserved word) cannot be used as variable names. Example, int, float, double, cout.
- There are two known naming conventions:
    - **Pascal case:** The first letter in the identifier and the first letter of each subsequent concatenated word are capitalized. For example: BackColor
    - **Camel case:** The first letter of an identifier is lowercase and the first letter of each subsequent concatenated word is capitalized. For example: backColor

Case-Sensitivity

- C++ is case-sensitive, which means that an identifier written in uppercase is not equivalent to another one with the same name in lowercase.
- For example, **myvariable** is not the same as **MYVARIABLE** and not the same as **MyVariable**.
- *These are three different variables.*
- Choose variable names that suggest the usage, for example: firstName, lastName.