Tutorial 7 - C++ Reference Variables & Typecasting

Topics Covered:

1. Built-in Data Types
2. Float, Double, and Long Double Literals
3. Reference Variables
4. Typecasting

---

1. Built-in Data Types

- Predefined by the C++ language and used directly.
- Example:

  ```
  int a, b, c; // Local variables
  c = a + b;   // Sum of a and b
  ```

- **Global Variables**: Declared outside all functions.
  - Accessed using the **Scope Resolution Operator (`::`)**.
  - Example:

    ```
    int c = 45; // Global variable
    cout << ::c; // Outputs the global variable c
    ```

- **Key Point**: Local variables take precedence over global variables within the same scope.

---

2. Float, Double, and Long Double Literals

- **Concept**:
  - By default, decimal literals (e.g., `34.4`) are treated as `double`.
  - To specify them as `float` or `long double`, use suffixes:
    - `F/f`: For `float`.
    - `L/l`: For `long double`.
- **Example Code**:

  ```
  float d = 34.4F; // Declared as float
  long double e = 34.4L; // Declared as long double

  cout << "Size of 34.4: " << sizeof(34.4) << endl;  // Default: double
  cout << "Size of 34.4F: " << sizeof(34.4F) << endl; // float
  cout << "Size of 34.4L: " << sizeof(34.4L) << endl; // long double
  ```

---

3. Reference Variables

- **Definition**: A reference variable is an alias for an existing variable.
- **Syntax**:

```
float x = 455;
float &y = x; // y is a reference to x
```

  - Any changes to `y` will affect `x` (and vice versa).

- **Example Output**:

```
cout << x; // Outputs: 455
cout << y; // Outputs: 455
```

---

4. Typecasting

- **Definition**: Converting a variable of one data type to another.
- **Syntax**:
  - C-style: `(type)variable` (e.g., `(float)a`).
  - Functional: `type(variable)` (e.g., `float(a)`).
- **Example Code**:

```
int a = 45;
float b = 45.46;

cout << (float)a << endl; // Converts int a to float
cout << int(b) << endl;   // Converts float b to int
```

- **Example Use in Expressions**:

```
cout << a + b << endl;       // Adds int and float
cout << a + (int)b << endl;  // Typecasts b to int, then adds
```

Code Example

```
#include <iostream>
using namespace std;

int c = 45; // Global variable

int main() {
    // Built-in Data Types
    int a = 5, b = 6, c;
    c = a + b; // Local variable
    cout << "Local c: " << c << endl;
    cout << "Global c: " << ::c << endl;

    // Float, Double, Long Double Literals
    float d = 34.4F;
    long double e = 34.4L;
    cout << "Size of 34.4: " << sizeof(34.4) << endl;
    cout << "Size of 34.4F: " << sizeof(34.4F) << endl;
    cout << "Size of 34.4L: " << sizeof(34.4L) << endl;

    // Reference Variables
    float x = 455;
    float &y = x;
    cout << "x: " << x << ", y: " << y << endl;

    // Typecasting
    int a = 45;
    float b = 45.46;
    cout << "Typecasted a: " << (float)a << endl;
    cout << "Typecasted b: " << int(b) << endl;
    cout << "Expression: " << a + (int)b << endl;

    return 0;
}

```

Short Notes

1. Built-in Data Types

- Predefined by C++.
- **Global Variables**: Declared outside functions, accessed using `::`.

2. Float, Double, Long Double Literals

- Default type for decimals: `double`.
- Suffixes:
  - `F/f` → Float
  - `L/l` → Long Double
- Example: `float d = 34.4F;`.

3. Reference Variables

- **Definition**: Alias for an existing variable.
- **Syntax**:

  ```
  float &ref = var;

  ```

4. Typecasting

- **Definition**: Converting one data type to another.
- **Methods**:
  - `(type)variable`
  - `type(variable)`