

## GECG10069 (561085) F25: Introduction to Programming (C++)

### Lab 2 : Literals & Variables



#### What you will learn from Lab 2

In this laboratory, you will further explore special output symbols and practice using variables in your programs.

#### TASK 2-1 : #DEFINE V.S. CONST NUMBER

- ✓ Figure out why coercion happened
- ✓ Try two different PI\_1 & PI\_2

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

#define PI 3.14159
// #define PI_1 3.14159    // 巨集常數
// #define PI_2 0.14159 + 3
const double PI_1 = 3.14159;
const double PI_2 = 3 + 0.14159;

int main() {

    int r = 5;
    double area = PI * r * r;    // 自動型別轉換 (int → double)
    double area1 = PI_1 * r * r; // 自動型別轉換 (int → double)
    double area2 = PI_2 * r * r; // 自動型別轉換 (int → double)
    // double area2 = PI_2 * static_cast<double>(r) * r; // 強制轉型

    cout << "Circle radius = " << r << ", Area = " << area1 << endl;
    cout << "Area with explicit casting = " << area2 << endl;

    return 0;
}
```

#### EXERCISE 2-1 : UNIFORMLY ACCELERATED MOTION

You're asked to write a C++ program that calculates the displacement of an object moving with uniform acceleration.

#### Formula

$$s = ut + \frac{1}{2}at^2$$

- $s$  = displacement (meters, double)
- $u$  = initial velocity (m/s, double)
- $a$  = acceleration (m/s<sup>2</sup>, double)
- $t$  = time (seconds, int)

## Input

- Initial velocity  $u$ : a double value in the range **0.0 to 100.0**
- Acceleration  $a$ : a double value in the range **-10.0 to 10.0**
- Time  $t$ : an int value in the range **1 to 100**

In this assignment, you do **not** need to read input from the user.  
Instead, define the values directly in your code as variables.

## Output Format

Initial Velocity: <value>

Acceleration: <value>

Time: <value>

Displacement: <value>

## Sample Output

Output:

Initial Velocity: 10

Acceleration: 2

Time: 5

Displacement: 75

Output:

Initial Velocity: 100

Acceleration: -9.8

Time: 3

Displacement: 255.9

Output:

Initial Velocity: 12.3

Acceleration: 4.7

Time: 7

Displacement: 201.25

## Sample Code:

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

int main() {
    // Define variables
    double u = 20.0; // initial velocity (0.0 ~ 100.0)
    double a = 0.0;  // acceleration (-10.0 ~ 10.0)
    int t = 8;       // time (1 ~ 100)

    // Compute displacement

    // Display

    return 0;
}
```

## EXERCISE 2-2 :WEIGHTED AVERAGE SCORE

You're asked to write a C++ program that calculates the weighted average score of four subjects:

Math, Physics, Chemistry, and English.

**Formula:**

$$\text{Weighted Average} = \frac{M \times w_M + P \times w_P + C \times w_C + E \times w_E}{w_M + w_P + w_C + w_E}$$

## Constant Weights

- Math : 1.0
- Physics : 1.5
- Chemistry : 2.5
- English : 3.3

## Input

- Four subject scores, each an integer in the range **0 to 100**.
- In this assignment, you do **not** need to read input from the user.  
Instead, define the scores directly in your code as variables.

## Output Format

Math: <value>

Physics: <value>

Chemistry: <value>

English: <value>

Weighted Score: <value>

## Sample Output:

Output:

Math: 70

Physics: 80

Chemistry: 90

English: 100

Weighted Score: 89.759

Output:

Math: 20

Physics: 40

Chemistry: 60

English: 80

Weighted Score: 59.5181

Output:

Math:0

Physics: 0

Chemistry: 0

English: 100

Weighted Score: 39.759

## Sample Code:

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

int main() {
    // Define constant weights
    const double WM = 1.0; // math weight
    const double WP = 1.5; // physics weight
    const double WC = 2.5; // chemistry weight
    const double WE = 3.3; // english weight

    // We will change scores here (0 ~ 100)
    int math = 20;
    int physics = 40;
    int chemistry = 60;
    int english = 80;

    // Compute weighted scores

    // Display

    return 0;
}
```