**📖 MTH2008** Scientific Computing Logbook

School of Engineering and Physical Sciences, University of Lincoln

Semester A, 2024

*“I confirm that this logbook is entirely my own work and that all references and quotations, from both primary and secondary sources, have been fully identified and properly acknowledged.”* - William Fayers (27378661).

**How well did I learn it?**

…

**How does my solution compare with the official solution?**

…

**How can I extend the concepts used in the tasks to form a new project**

…

**Lesson Review Question Answers**

…

### Basic Use: Variables And Data Types

#### Task 1.2

Declare and initalise the variables

int total\_children = 10;  
int candies\_per\_child = 5;  
int extra\_candies = 10;

Output the starting values and the total number of candies

std::cout << "HALLOWEEN CANDY CALCULATOR" << std::endl;  
std::cout << "Total Children: " << total\_children << std::endl;  
std::cout << "Candies per Child: " << candies\_per\_child << std::endl;  
std::cout << "Extra Candies: " << extra\_candies << std::endl;  
std::cout << "---" << std::endl;  
std::cout << "Total Candies: " << total\_candies << std::endl;

Return 0 to indicate the program ran successfully

return 0;

#### Task 1.2

unsigned short int can be broken down into three parts:- unsigned: This means the variable can only store positive values, which fits the context of candies, as we cannot have a negative number of candies.- short: This means the variable can store values between 0 and 65,535, which is more than enough for the number of candies we are calculating.- int: This means the variable is an integer, which is the type of data we are working with.Hence, unsigned short int is the most appropriate data type for the variable total\_candies.

unsigned short int total\_candies;

Calculate the total number of candies by multipling the average per child by the total number of children, then adding the extra candies.

total\_candies = total\_children \* candies\_per\_child + extra\_candies;

### Basic Use: Input And Math

#### Task 2.2

To calculate the remainder of two floats without using any library functions, we can cast the result of the division to an integer, multiply it by the second number, and subtract it from the first number. This will give us the remainder of the division.

result = number1 - (static\_cast<int>(number1 / number2) \* number2);  
std::cout << "Remainder: " << result << std::endl;

#### Task 2.3

We can create a function called calculate\_gravitational\_force that takes the masses of two objects and the distance between them as input parameters. The function calculates the gravitational force between the two objects using the formula:

float calculate\_gravitational\_force(float mass1, float mass2, float distance) {  
const float G = 6.67430e-11;  
if (distance == 0) {  
throw std::invalid\_argument("Distance cannot be zero.");  
}  
return (G \* mass1 \* mass2) / (distance \* distance);  
}

#### Task 2.3

Declare variables to store the masses and distance.

float mass1, mass2, distance;

We then input the masses and distance.

std::cout << "Enter the mass of the first object (kg): ";  
std::cin >> mass1;  
std::cout << "Enter the mass of the second object (kg): ";  
std::cin >> mass2;  
std::cout << "Enter the distance between the objects (m): ";  
std::cin >> distance;

And finally we call the function, catching any exceptions that occur.

try {  
float force = calculate\_gravitational\_force(mass1, mass2, distance);  
std::cout << "Gravitational Force: " << force << " N" << std::endl;  
} catch (std::invalid\_argument& error) {  
std::cerr << "Error: " << error.what() << std::endl;  
}

## 📓 References

[1] “A comprehensive guide to C++ coding standards and best practices,” *C++ Style Guide*, 2024. [Online] Available: https://google.github.io/styleguide/cppguide.html (accessed 2024-10-05).

[2] “Official documentation for Doxygen’s documentation blocks and usage,” *Doxygen Manual - Documentation Blocks*, 2024. [Online] Available: https://www.doxygen.nl/manual/docblocks.html (accessed 2024-10-05).