# Lab 1 – Tasks Solutions

# Solution for Task 1: Declaring and Initializing Variables

#include <iostream>  
#include <string>  
  
int main() {  
 // Declare and initialize variables  
 int age = 20;  
 float height = 1.65;  
 char grade = 'B';  
 std::string name = "Alice";  
  
 // Display initial values  
 std::cout << "Initial values:" << std::endl;  
 std::cout << "Name: " << name << ", Age: " << age << ", Height: " << height << ", Grade: " << grade << std::endl;  
  
 // Update variables  
 name = "Bob";  
 age = 22;  
 height = 1.75;  
 grade = 'A';  
  
 // Display updated values  
 std::cout << "Updated values:" << std::endl;  
 std::cout << "Name: " << name << ", Age: " << age << ", Height: " << height << ", Grade: " << grade << std::endl;  
  
 return 0;  
}

# Solution for Task 2: Arithmetic Operations with User Input

#include <iostream>  
  
int main() {  
 int num1, num2;  
  
 // Input two integers from the user  
 std::cout << "Enter the first number: ";  
 std::cin >> num1;  
 std::cout << "Enter the second number: ";  
 std::cin >> num2;  
  
 // Perform arithmetic operations  
 int sum = num1 + num2;  
 int difference = num1 - num2;  
 int product = num1 \* num2;  
 float quotient = static\_cast<float>(num1) / num2; // Type casting for float division  
 int remainder = num1 % num2;  
  
 // Display the results  
 std::cout << "Addition: " << sum << std::endl;  
 std::cout << "Subtraction: " << difference << std::endl;  
 std::cout << "Multiplication: " << product << std::endl;  
 std::cout << "Division: " << quotient << std::endl;  
 std::cout << "Modulus: " << remainder << std::endl;  
  
 return 0;  
}

# Solution for Task 3: Variable Scope and Lifetime

#include <iostream>  
  
// Global variable  
int globalVar = 0;  
  
void demoFunction() {  
 // Local variable  
 int localVar = 0;  
  
 // Static variable  
 static int staticVar = 0;  
  
 // Increment all variables  
 localVar++;  
 staticVar++;  
 globalVar++;  
  
 // Display the values  
 std::cout << "Local: " << localVar << ", Static: " << staticVar << ", Global: " << globalVar << std::endl;  
}  
  
int main() {  
 // Call the function multiple times  
 demoFunction();  
 demoFunction();  
 demoFunction();  
  
 return 0;  
}

# Solution for Task 4: Type Casting and Conversion

#include <iostream>  
  
int main() {  
 int num1, num2;  
  
 // Input two integers from the user  
 std::cout << "Enter the first number: ";  
 std::cin >> num1;  
 std::cout << "Enter the second number: ";  
 std::cin >> num2;  
  
 // Implicit type casting (integer division)  
 int resultWithoutCasting = num1 / num2;  
 std::cout << "Result without type casting: " << resultWithoutCasting << std::endl;  
  
 // Explicit type casting (float division)  
 float resultWithCasting = static\_cast<float>(num1) / num2;  
 std::cout << "Result with type casting: " << resultWithCasting << std::endl;  
  
 return 0;  
}

# Solution for Task 5: Complex Arithmetic Expression and Precedence

#include <iostream>  
  
int main() {  
 // Without parentheses  
 int resultWithoutParentheses = 5 + 3 \* 2 / 4 - 1;  
 std::cout << "Result without parentheses: " << resultWithoutParentheses << std::endl;  
  
 // With parentheses  
 float resultWithParentheses = (5 + 3) \* (2 / (4 - 1));  
 std::cout << "Result with parentheses: " << resultWithParentheses << std::endl;  
  
 return 0;  
}