**COSC 1437 Programming Fundamentals II**

**By KariAnn Harjo**

16 Exceptions & templates

Exercise 1:

Write templates for the two functions minimum and maximum. The minimum function should accept two arguments and return the value of the argument that is the lesser of the two. The maximum function should accept two arguments and return value of the argument that is the greater of the two. Design a simple driver program that demonstrates the templates with various types.



**#include <iostream>**

**#include <string>**

**using namespace std;**

**// Function template for finding the minimum value**

**template<typename T>**

**T findMinimum(T a, T b) {**

**return (a < b) ? a : b;**

**}**

**// Function template for finding the maximum value**

**template<typename T>**

**T findMaximum(T a, T b) {**

**return (a > b) ? a : b;**

**}**

**int main() {**

**// Example usage with integer values**

**int num1 = 3;**

**int num2 = 5;**

**int minInt = findMinimum(num1, num2);**

**int maxInt = findMaximum(num1, num2);**

**cout << "Minimum of " << num1 << " and " << num2 << ": " << minInt << endl;**

**cout << "Maximum of " << num1 << " and " << num2 << ": " << maxInt << endl;**

**// Example usage with floating-point values**

**double num3 = 3.5;**

**double num4 = 5.5;**

**double minDouble = findMinimum(num3, num4);**

**double maxDouble = findMaximum(num3, num4);**

**cout << "Minimum of " << num3 << " and " << num4 << " is: " << minDouble << endl;**

**cout << "Maximum of " << num3 << " and " << num4 << " is: " << maxDouble << endl;**

**// Example usage with strings**

**string str1 = "hello";**

**string str2 = "hi";**

**string minString = findMinimum(str1, str2);**

**string maxString = findMaximum(str1, str2);**

**cout << "Minimum string of \"" << str1 << "\" and \"" << str2 << "\" is: " << minString << endl;**

**cout << "Maximum string of \"" << str1 << "\" and \"" << str2 << "\" is: " << maxString << endl;**

**return 0;**

**}**



Exercise 2:

Write a function template that accepts an argument and returns its absolute value. The absolute value of a number is its value with no sign. For example, the absolute value of -7 is 7, and the absolute value of 6 is 6. Test the template in a single driver program.



#include <iostream>

using namespace std;

// Function template for calculating the absolute value

template<typename T>

T calculateAbsoluteValue(T num) {

return (num < 0) ? -num : num;

}

int main() {

// Test the calculateAbsoluteValue function template with different types of numbers

// Test with integers

int num1 = -5;

int absValue1 = calculateAbsoluteValue<int>(num1);

cout << "Absolute value of " << num1 << ": " << absValue1 << endl;

int num2 = 4;

int absValue2 = calculateAbsoluteValue<int>(num2);

cout << "Absolute value of " << num2 << ": " << absValue2 << endl;

// Test with floating-point numbers

double num3 = -5.5;

double absValue3 = calculateAbsoluteValue<double>(num3);

cout << "Absolute value of " << num3 << ": " << absValue3 << endl;

double num4 = 4.5;

double absValue4 = calculateAbsoluteValue<double>(num4);

cout << "Absolute value of " << num4 << ": " << absValue4 << endl;

return 0;

}  
  
