# Northeastern University

ETC 1103 – Test #2

A number of the form a + ib in which a and b are real numbers and i2 = -1, is called a Complex number. The number a is called the *real part* and b is called the *imaginary part*. Complex numbers can also be represented as ordered pair (a, b). Addition and multiplication of complex numbers are defined by the following rules: **[50]**

(a + ib) + (c + id) = (a+c) + i(b + d) // Addition

(a + ib) \* (c + id) = (ac – bd) + i(ad + bc) // Multiplication

Using the ordered pair notation, these rules are written as:

(a, b) + (c, d) = ((a + c), (b + d))

(a, b) \* (c, d) = ((ac – bd), (ad + bc))

C++ has no built- in data type that allows us to manipulate complex numbers. In this assignment, you will define a data type, complexType, which can be used to process complex numbers. Define constructors to define complex numbers of the form **a, a+ib,** and **ib**. Overload the **+** and \* arithmetic operators, the equality operator, the assignment operator, and the insertion and extraction operators **<<** and **>>**.

Use the complexType in a program given below.

#include “complexType.h”

int main(void)

{

complexType n1(23, 34);

complexType n2, n3;

complexType n4;

cout << “n1 = “ << n1 << endl;

cout << “n2 = “ << n2 << endl;

cout << “Enter a complex number in the form (a, b): “;

cin >> num2;

cout << endl;

cout << “New value of n2 = “ << n2 << endl;

n4 = n2;

cout << “n4 must have the same value as n2: “ << n4 << endl;

n3 = n1 + n2;

cout << “n3 = n1 + n2 is “ << n3 << endl;

if (n4 == n2)

cout << “n4 is equal to n2.” << endl;

else

cout << “n4 and n2 are different!” << endl;

// Test Multiply operator

cout << n1 << “ \* ” << n2 << “ = “ << n1 \* n2 << endl

return 0;

}