2021 Winter CIS200 – Programming Assignment 2

List, Operator and Friend

Release date: Feb 16, 2021

Due date: Mar 04, 2021

# **Question 1 Unsorted List (30 points)**

Implement a template class UnsortedList as defined by the following skeleton:

#define MAX\_ITEMS 10

typedef char ItemType;

class UnsortedList

{

private:

int length;

ItemType values[MAX\_ITEMS];

int currentPos;

public:

UnSortedList( ); // default constructor: lenght=0, currentPos=-1

void MakeEmpty; // let length=0

void InsertItem(ItemType x); // insert x into the list

void DeleteItem(ItemType x); // delete x from the list

bool IsFull( ); // test if the list is full

int Lengthls( ); // return length

void RetrieveItem(ItemType &x, bool &found); // retrieve x from the list, the

// boolean result is stored in found

void ResetList( ); // currentPos=-1

void GetNextItem(ItemType &x); // get the next element from the list with

// respect to the currentPos

void printElement( ); // print out the values of all the list elements, separated by

// a white space

};

You need to use linear search in InsertItem( ), DeleteItem( ) and RetrieveItem( ). You should create one instances of this class and read in data from a file: char.dat, which can be downloaded from cis 200 web site and are assumed to be in the root directory of your project. Data in char.dat contains chars, which should be inserted into the object of UnsortedList. Note the you do not have any prior knowledge about how data values in char.dat.

You also need to write a subroutine, int numCharElement(ifstream &x), to determine the number of char elements in char.dat.

The skeleton of main( ) should be:

Int main( )

{

UnsortedList x;

// open char.dat

…… // set up x based on the data in char.dat

x.printElement( );

cout << numCharElement( ) << endl;

return 0;

}

# **Question 2 Operator (30 points)**

Define a class for complex numbers. A complex number is a number of the form: a + b \* i, where a and b are number of type double, and i is a number that represents . Represent a complex number as two values of type double. Name the number variables *real* and *imaginary*. Call the class Complex. Include a constructor with two parameters of type double that can be used to set the member variables of an object to any values. Include a constructor that has only a single parameter of type double; call this parameter realPart and define the constructor so that the object will be initialized to realPart + 0 \* i. Include a default constructor that initializes an object to 0 + 0 \* i. Overload all the following operators so that they correctly apply to the type Complex class:

1. = = equivalence operator (test if two complex numbers are equal to each other)
2. +
3. –
4. \*
5. /
6. = assignment operator
7. << output the content of real and imaginary part to computer screen
8. ++ (prefix operator: add both real and imaginary parts by 1)
9. ++ (postfix operator: add both real and imaginary parts by 1)
10. You need to implement a copy constructor

To add or subtract two complex numbers, add or subtract the two member variables of type double. The product of two complex numbers is given by the following formula:

(x + y \* i) \* (u+v \* i) = (x\*u – y\*v) + (x\*v+y\*u) \* i

The division of two complex numbers is defined by

In your main( ) function, you should perform the following tests:

Complex x(2, 2);

Complex y(4, 3);

Complex z, w, v, q;

z = x + y;

w = x – y;

v = x \* y;

q = x / y;

if( z = = w)

cout << “ z = w” << endl;

else

cout << “ z != w” << endl;

// print out the real and imaginary parts of z, w, v, and q

…..

In order to print out the values of complex numbers, you may design a public member function print ( ) to do so.

You also need to design test cases for ++, <<, and copy constructor.