

# PS # 7

**Deadline: 09.05.2020 at 23:50**

## Hand-in Policy:

Create a cpp file for each part of the assignment and save the screenshot while these parts are running.

Your assignment should consist of 3 cpp files and 3 screenshots.

Zip all files of the assignment. Your zip files should look like this

Ps7\_studentnumber\_name\_surname.zip

**1)**

Write a program that accepts a C-string input from the user and reverses the contents of the string. Your program should work by using two pointers. The “head” pointer should be set to the address of the first character in the string, and the “tail” pointer should be set to the address of the last character in the string (i.e., the character before the terminating null ). The program should swap the characters referenced by these pointers, increment “head” to point to the next character, decrement “tail” to point to the second-to-last character, and so on, until all characters have been swapped and the entire string reversed.

**2)**

Create a class named Student that has three member variables:

name – A string that stores the name of the student

numClasses – An integer that tracks how many courses the student is currently enrolled in

classList – A dynamic array of strings used to store the names of the classes that the student is enrolled in

Write appropriate constructor(s), mutator, and accessor functions for the class along with the following:

- A function that inputs all values from the user, including the list of class names. This function will have to support input for an arbitrary number of classes.
- A function that outputs the name and list of all courses.
- A function that resets the number of classes to 0 and the classList to an empty list.
- An overloaded assignment operator that correctly makes a new copy of the list of courses.
- A destructor that releases all memory that has been allocated.

Write a main function that tests all of your functions.

**3)**

Using dynamic arrays, implement a polynomial class with polynomial addition, subtraction, and multiplication.

Discussion: A variable in a polynomial does nothing but act as a placeholder for the coefficients. Hence, the only interesting thing about polynomials is the array of coefficients and the corresponding exponent. Think about the polynomial

$x^3 + x + 1$

Where is the term in  $x^*x$  ? One simple way to implement the polynomial class is to use an array of doubles to store the coefficients. The index of the array is the exponent of the corresponding term. If a term is missing, then it simply has a zero coefficient.

There are techniques for representing polynomials of high degree with many missing terms. These use so-called sparse matrix techniques. Unless you already know these techniques, or learn very quickly, do not use these techniques.

Provide a default constructor, a copy constructor, and a parameterized constructor that enables an arbitrary polynomial to be constructed.

Supply an overloaded operator  $=$  and a destructor.

Provide these operations:

polynomial + polynomial, constant + polynomial, polynomial + constant,

polynomial - polynomial, constant - polynomial, polynomial - constant.

polynomial \* polynomial, constant \* polynomial, polynomial \* constant,

Supply functions to assign and extract coefficients, indexed by exponent.

Supply a function to evaluate the polynomial at a value of type double .

You should decide whether to implement these functions as members, friends, or standalone functions.