NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES ISLAMABAD CAMPUS

CS-1004 Object Oriented Programming Spring-2024
ASSIGNMENT-02
Sections (All)

Deadline: 25th March, 2024

Instructions:

1) Please start early otherwise you will struggle with the assignment.
2) Assignments are to be done individually. The code you write must be your own and you must understand each part of your code. You are encouraged to get help from the instructional staff through google classroom.
3) Do not use any String or math libraries (such as cmath, cstring, string etc) and do not use built-in function (such as strlen, strcmp etc) except in Q3 (where built-in string is allowed). Caution: zero marks will be awarded.
4) Your code must be generic. Do not change Function Prototypes. Caution: zero marks may be awarded.
5) Test Cases are provided for each question. Your code will be evaluated with similar test cases. If the required output is generated, you will be awarded full marks. Failing to generate the correct output will result in zero marks. **Total Marks: 200.**
6) Marks are also allocated for good programming practices like indenting code, commenting, resource allocation (freeing dynamic memory) etc.
7) Plagiarism of any kind (copying from others, copying from the internet, etc) is not allowed. If found plagiarized, you will be awarded zero marks in the whole assignment category.
8) Test cases: Test cases (in gtest) will be shared with you on Google Classroom. We will be running your code against our test cases, and a test case failure or a segmentation fault/incorrect result or even syntax error will result in zero marks.
9) **Submission Guidelines**: Dear students, we will be using auto-grading tools (gtest), so failure to submit according to the below format would result in zero marks in the relevant evaluation instrument.

    a) Your folder should contain **4 .cpp** files and **3 .h** files.
    b) Please don't include the main function while submitting the files for Q1 and Q2. The files must contain your name, student-id, and assignment # on the top of the file in the comments.
    c) Rename the folder as ROLL-NUM_SECTION (e.g. 23i-0001_A) and compress the folder as a zip file. (e.g. 23i-0001_A.zip). Only zip file will be acceptable.
    d) Names of the individual files should be as follows: ROLL-NUM_SECTION_Q1.cpp, ROLL-NUM_SECTION_Q1.h and so on. Make Sure to include the main.cpp of Q3.
    e) Submit the zip file on google forms. If we are unable to download your submission due to any reason, you will be awarded zero mark.
    f) Start the submission process well before time so that you can overcome problems as you face them.

Note: Follow the given instructions to the letter, failing to do so may result in a zero.

NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES ISLAMABAD CAMPUS

Question # 1 **[50 Marks]**

In a particular stationery store, to hold the article and container information, Container structure is used. The shopkeeper wants to maintain a chain of article containers on so that it's easier to perform different operations. Implement the Shop Class member functions, please use the correct input parameters and return types.

```
struct Container
{
  char *name;
int containerno;
  Container* link;
}
class Shop {
Public:
    Container* start;
    // Default constructor
    Shop() { start = NULL; }

    //Copy Constructor
    Shop(const Shop& other);


void add_Container( char *name,int containerindex);  /*This member function will
take the name and containerindex of Article as input parameter. It creates the
Container instance. If the new Container instance is the first in the chain then
the pointer "start" will point to it. Otherwise a new instance is attached at the
end of the existing chain.*/

void print_Shop(); //It prints all the articles with their container number in
the chain.

void delete_Chain( int containerindex); /* This member function will take
the containerindex of Article as input parameter. It deletes the
mentioned containerindex from the chain of Shop instance. And rejoins the
remaining chain.*/

void Sort_Chain(); //It sort all the elements in the chain based upon their
containerindex.

void update_name_at_containernumber(int containerindex, char * name);

 void remove_Duplicate(); //It removes all the containers with same
article in the chain.

void findContainer (int containerindex);// It prints article name in the selected
container.

void findContainer (int containerindex1, int containerindex2);// It prints
article name in the container number range. The range is inclusive

~Shop()

};
```

NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES ISLAMABAD CAMPUS

Question # 2 **[50 Marks]**

Your goal is implementing "String" class with the following functions. You will need to write two files (String.h and String.cpp). Your implemented class must fully provide the definitions of following class (interface) functions. For naming the files, follow the convention mentioned at the beginning.

```
class String {
 // think about the private data members
public:
// provide definitions of following functions
String(); // default constructor
String(const char *str); // initializes the string with constant c-string
String(const String &); // copy constructor to initialize the string from the existing
string
String(int x); // initializes a string of predefined size

char* getdata(); //returns the string inside the object

// Binary Operators //
Sub-script Operators
const char operator[](int i) const; // returns the character at index [x]
//NOTE: in above [] operator functions if i=negative int value, print ith character
from end //of string e.g. in case of "LOOP" if i=-1 OR i=3, it should return 'P'
similarly i=-4 OR i=0, //return 'L'

// Arithmetic Operators
String operator+(const String &str); // appends a String at the end of the String
String operator+(const char &str); // appends a char at the end of the String
String operator+(const char *str); // appends a String at the end of the String
String operator-(const String &substr); //removes the substr from the String
String operator-(const char &str); //removes all occurrences of char from the String
String operator-(const char* str); //removes the str from the String

// Assignment Operators
String& operator=(const String&); // copies one String to another
String& operator=(char*); // copies one c-string to another

// Logical Operators
bool operator==(const String&) const; // returns true if two Strings are equal
bool operator==(const char *) const; // returns true if the c-string is equal to the
String

// Unary Operators //
Boolean Not Operator
bool operator!(); // returns true if the String is empty

// Function-Call Operators
//If something is not found then return -1
int operator()(char) const; // returns the first index of the character being
searched
int operator()(const String&) const; // returns the first index of the String
being searched
int operator()(const char*) const; // returns the index of the c-string being
searched
```

```
// Conversion Operator
operator int() const; // returns the length of string
~String(); // destructor
};
ostream& operator<<(ostream& output, const String& str); // outputs the string
istream& operator>>(istream& input, String& str); // inputs the string
```

Question # 3 **[100 Marks]**

Consider the following figure

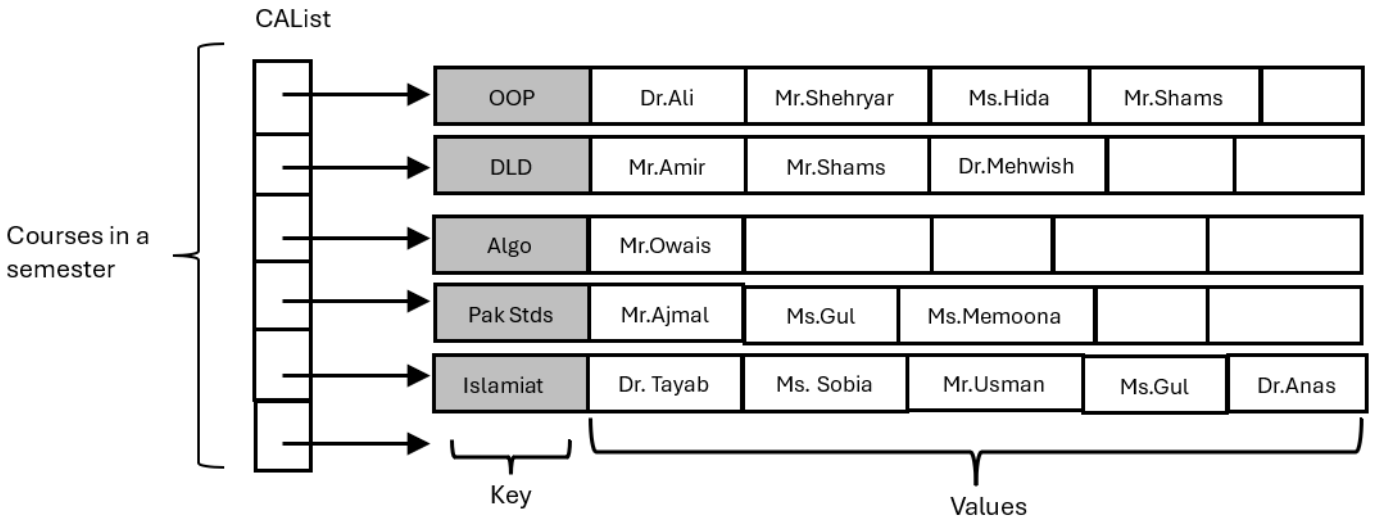

Figure 1: tt3 For reference

We need a Course Allocation List to store data about course instructors. A CAList consists of multiple [key, values] entries, where space is dynamically allocated when needed. Typically, multiple values may be mapped against one key. For simplicity, we assume that maximum six (06) values can be stored against one key (as shown in figure). And CAList cannot have more than 7 Keys ***(No key occurs more than once in CAList)***. Here key is the specialty of course instructor and values are names of teachers. You are required to design and implement CAList class (and any other required classes) in C++, where following code should run without any syntax/logical errors :

- an overloaded += operator
- an overloaded -= operator

Operator -= will delete the required name and shift the remaining elements to the left

```
int main()
{
    CAList tt, tt2, tt3; // assume CAList is the name of class
    // operator[] is used to get "entry" against given "key" from CAList and
```

```
    // can be used in following context with operator= to create/refresh an
entry
    tt["OOP"] = "Dr.Ali";
    tt["OOP"] = "Mr.Shehryar";
    tt["OOP"] = "Ms.Hida";
    tt["OOP"] = "Mr.Shams";
    tt["DLD"] = "Mr.Amir";
    tt["DLD"] = "Mr.Shams";
    tt["DLD"] = "Dr.Mehwish";

      cout<<tt;      //output should look like:  [ OOP : { Dr.Ali,
Mr.Shehryar ,Ms.Hida,Mr.Shams},
                                                  DLD : { Mr.Amir,Mr.Shams,Dr.Mehwish } ]



    // operator+= with an "entry" is used to add new "value" against a given "key".
    // if entry with given "key" does not exist then new entry is created in
CAList.
    // Following operators += and = perform the same operation

    tt2["Algo"] = "Mr.Owais";
    tt2["OOP"] = "Mr.Shehryar";
    tt2["Pak Stds"] = "Mr.Ajmal";
    tt2["Pak Stds"] += "Ms.Gul";
    tt2["Pak Stds"] = "Ms.Memoona";
    tt2["Islamiat"] = "Dr.Tayab";
    tt2["Islamiat"] += "Ms.Sobia";
    tt2["Islamiat"] = "Mr.Usman";
    tt2["Islamiat"] += "Ms.Gul";
    tt2["Islamiat"] += "Mr.Anas";

    // following statement creates a new entry in CAList, and then adds
values to it
    // operator+ is also used to add two (CAList)s and return a new CAList.
    // operator= is used to assign one List to another(deep copy).
     tt3 = tt + tt2; // tt3 now looks like the CAList in figure
above! (same as set union operation)

  /* At this stage tt, tt2, and tt3 look like following  respectively:

tt:

                                [ OOP : { Dr.Ali, Mr.Shehryar,Ms.Hida,Mr.Shams},

                                DLD : { Mr.Amir,Mr.Shams,Dr.Mehwish } ]


tt2:

                                [ Algo : { Mr.Owais },
```

NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES ISLAMABAD CAMPUS

OOP : {Mr.Shehryar },
Pak Stds : { Mr.Ajmal,Ms.Gul,Ms.Memoona} ,
Islamiat : { Dr.Tayab,Ms.Sobia,Mr.Usman,Ms.Gul,Mr.Anas }]

tt3:

[OOP : { Dr.Ali, Mr.Shehryar,Ms.Hida,Mr.Shams},
DLD : { Mr.Amir,Mr.Shams,Dr.Mehwish },
Algo : {Mr.Owais },
Pak Stds : { Mr.Ajmal,Ms.Gul,Ms.Memoona} ,
Islamiat : { Dr.Tayab,Ms.Sobia,Mr.Usman,Ms.Gul,Mr.Anas }]/

```
	tt4 = tt - tt2; //This Operation will work similar to the set difference operation
```

tt4:

[OOP : { Dr.Ali,Ms.Hida,Mr.Shams},
DLD : { Mr.Amir,Mr.Shams,Dr.Mehwish }]/

```
	return 0;
}
```

(built-in string is allowed for Q3 ONLY!!!