CS-1004 Object Oriented Programming Spring-2023
ASSIGNMENT-02
Section (All)

Submission Deadline: 20th March, 12:00 pm.

Instructions:

1. Assignments are to be done individually. You must complete this assignment by yourself. You cannot work with anyone else in the class or with someone outside of the class. 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.
2. Do not use any String or math libraries (such as cmath, cstring, string etc) and also do not use built-in function (such as strlen, strcmp etc) unless specified. **Caution**: zero marks may be awarded.
3. Do not edit **Function Prototypes**. **Caution**: zero marks may be awarded.
4. The usage of string is strictly prohibited.
5. Your code must be **generic , use dynamically created arrays only**.
6. Marks distribution and 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 may result in zero marks.
7. **Plagiarism**: 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 assignment. Repeating such an act can lead to strict disciplinary actions and failure in the course.
8. **Please start early otherwise you will struggle with the assignment.**
9. **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.**
10. **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. Make a folder titled **21X-XXXX_A2_OOP** and put all your .cpp files and .h files in it.
    b. Your folder should contain **4 .cpp** files and **4 .h** files.
    c. You will zip the folder and submit it on google forms.
    d. No other method of submission will be accepted.
    e. 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.

Questions

Question # 1 30 marks

In a particular Zoo, to hold the zoo cage information ZooCage structure is used. The Zookeeper wants to maintain a chain of zoo cages on so that it’s easier to perform different operations. Implement the ZooChain Class member functions, please use the correct input parameters and return types.

```
struct ZooCage
{
  char *name;
  int cageNumber;
  ZooCage* link;
}

class ZooChain {
    ZooCage* start;

public:
    // Default constructor
    ZooChain() { start = NULL; }


void add_Cage( char *name,int cageNumber);  /*This member function will take the
name and cageNumber of Animal as input parameter. It creates the ZooCage
instance. If the new Zoocage 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_Chain(); //It prints all the animals with their cage number in the
chain.
void delete_Chain( int cageNumber); /* This member function will take the
cageNumber of Animal as input parameter. It deletes the mentioned cageNumber from
the chain of  ZooCage instances.And rejoins the remaining chain.*/
void Sort_Chain();// It sort all the elements in the chain based upon their
cageNumber.
void update_name_at_cageNumber(int cageNumber, char * name)

void remove_Duplicate();// It removes all the duplicates in the chain.

~ZooChain()

};
```

Question # 2 String **40 marks**

Your goal is implement "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.

```
 class String {
 char* data;
 // there is need of more data members, think about them...
 public:
 // provide definitions of following functions...
 String(); // default constructor
 String(char* str); // constructor9 String(char *str); // initializes the
string with constant cstring
 String(const String &); // copy constructor to initialize the string from
existing string
 String(int x); // initializes a string of pre-defined size
 char getChar(int i); // returns the character at index [x] in a string

 bool isEmpty(); // returns true if string is empty..

 String& copy_String(const String&); // Copy one string to another ...

String append_string(const String &str) const; // appends a String at the
end of the String
String append_string(const char &str) const; // appends a char at the end of
the String
String append_string(char *&str) const; // appends a String at the end of
the String
String append_string(string& str) const; // appends a String at the end of
the String


String remove_string(const String &substr) const; //removes the substr from
the String
String remove_string(const string &substr) const; //removes the substr from
the String



 String& assign_string(const String&); // copies one String to another
 String& assign_string(char*); // copies one c-string to another
 String& assign_string(const string&); // copies one string to another


 bool isEqual(const String&)const; //returns true if two Strings are equal
 bool isEqual(const string&)const; //returns true if two strings are equal
 bool isEqual(char *)const; //returns true if two strings are equal

int index_At (char) const; // returns the index of the character being
searched
int index_At (const String&) const; // returns the start index of the String
being searched
int index_At (const string&) const; // returns the start index of the string
being searched
int index_At (char *) const; // returns the start index of the c-string
being searched

String mul_String(int i); //multiples the string by i times and return the
string.
int length(); // returns the length of string
 ~String(); // destructor...  };
```

Question # 3

BinaryStore Calculator **100 marks**

A BinaryStore calculator will store bytes “stored in strings” with their addresses “stored in strings”, i.e., at each address in the BinaryStore there is a stored byte.

Each address will be 4 characters string and each byte will be 8 characters strings.
Your task is to create the BinaryStore class with all the function given in main class. Please use the correct input arguments and return type for the function of BinaryStore class.

Furthermore, implement all the constructors and member functions required against following sample code

```
class BinaryStore
{
// add suitable member attributes


public:

BinaryStore (int len); // constructor which creates the store of length len.

void add_Address(char*  s); // add a new address in the list of Binary store.

void set_Byte(char*  address, char* value); // add the byte at newly added address
// here the first parameter is address and second is byte// Similarly the
following code adds byte address and later byte // if address not found add
address and byte both // function also overrides the data given at particular byte
address.
char* Get(char*  address),; //Get function will retrieve the byte at a given
address ,

char* ToString(); //return the BinaryStore as character dynamic array.

char * Add(char * a, char * b);

//ADD function will take two strings as input parameters and given the binary
equivalent in form of string.


// comp_EQUAL, comp_AND, and  comp_OR function will take two strings as input
parameters and given the binary equivalent in form of string.


bool      comp_EQUAL(char * a, char * b);
char *    comp_AND(char * a, char * b);
char *    comp_OR(char * a, char * b );
char *    comp_NOT(char * a, char * b);

~BinaryStore()

};
```

Question # 4 **30 marks**

We have lots data of students containing their name, addresses, age and CGPA. We want to create a class Record that will help to read and write this data to file. Class would provide member functions that will ease the data access and retrieval.

```
class Record
 {
       private :

     // add required data members if required
       char* fileName;


       public :

  Record (char* str) ; constructor will create the file with the given file name.

  int  record_Input(); /// input function will get the multiple data (name ,address
,age and CGPA )from the user and adds to the file,
  void record_Display() ;// the function will display the all the data store in
file.
  int  delete_Record();/// function will delete the complete record that is data
(name ,address ,age and CGPA ) of person with given name.
  int  record_Count();//function will count the total records stored in the file.
  int  record_Search(); // Allows to search the file with a particular name
  int  record_Insert();//it will allow the user to insert the data (name ,address
,age and CGPA ) after some name in the file
  void record_Replace();//it will allow the user to replace the data (name ,address
,age and CGPA ) of given name in the file

~Record()

 } ;
```

Happy Coding 😊