CS 103 Computer Programming

Assignment Number 3

March 16, 2017

Deadline: Thursday 23 March, 2017 before 19h30

Attention

- Make sure that you read and understand each and every instruction. If you have any questions or comments you are encouraged to discuss your problems with your colleagues (and instructors) on Piazza.
- Plagiarism is strongly forbidden and will be very strongly punished. If we find that you have copied from someone else or someone else has copied from you (with or without your knowledge) both of you will be punished. You will be awarded straight zero in this assignment or all assignments.
- Submit three files "*.h, *.cpp and main.cpp" file for each question of your assignment.

Q1: Implementation of String Class Your goal is to implement a generic "String" class using cstring, *i.e.*you will need to write three files (string.h, string.cpp and stringMain.cpp). Your implemented class must fully provide the definitions of following class (interface) functions . Please also write down the test code to drive your class implementation. **Please note that we will be running your code against our test code and any segmentation faults or incorrect result will result in loss of marks.**

```
class String{
// think about the private data members...
public:
// provide definitions of following functions...
String();// default constructor
String(char *str);// initializes the string with constant cstring
String(const String &);// copy constructor to initialize the string fromexisting
  ↪ string
String(int x);// initializes a string of pre-defined size
char getAt(int i);// returns the character at index [x]
void setAt(int i, char c);// set the character at index [x]
String substr(int pos, int len);// returns a substring of length len startingfrom
  ↪ location 'pos'
String substr(int pos);// returns substring from the given position to the end.
void append(char a);// append a char at the end of string
void append(String str );// append a String at the end of string
void append(char *str );// append a constant c string at the end of string
int length();// returns the length of string
char * tocstring();// converts a String to c-string
void display();// displays the string ..
bool isEmpty();// returns true if string is empty..
void copy(const String&);// Copy one string to another ...
void copy(const char *);// copy cstring to String...
int find(char);// returns the index of character being searched.
bool equal(String);// should return true if both strings are same
int stoi();// function for converting a string to integer.
void split(char token, string *&, int &ntokens); // should split the stringaccording
  ↪ to given token and store all substrings in a dynamic arraypassed as second
  ↪ argument, and return ntokens in third argument.
bool isanagram(String &);// return true if the given string is anagram of theinput
  ↪ string, an string is called anagram of another string if it containsexactly same
  ↪ characters but might be different order of appearance.
~String();// destructor...
};
```

Q2: Implementation of Array Class Your goal is to implement a generic "Array" class using cstring, *i.e.*you will need to write three files (array.h, array.cpp and arrayMain.cpp). Your implemented class must fully provide the definitions of following class (interface) functions . Please also write down the test code to drive your class implementation. **Please note that we will be running your code against our test code and any segmentation faults or incorrect result will result in loss of marks.**

```cpp
class Array{
// think about the private data members...
public:
// provide definitions of following functions...
Array();// a default constructor
Array(int size);// a parametrized constructor initializing an Array of predefined
  ↪ size
Array(int *arr, int size);// initializes the Array with an existing Array
Array(const Array &);// copy constructor
int getAt(int i);// returns the integer at index [i]
void setAt(int i, int val);// set the value at index [i]
Array subArr(int pos, int siz);// returns a sub-Array of size siz starting from
  ↪ location 'pos'
Array subArr(int pos);// returns a sub-Array from the given position to the end.
int * subArrPointer(int pos, int siz);// returns an array of size siz starting from
  ↪ location 'pos'
int * subArrPointer(int pos);// returns an array from the given position to the end.
void push_back(int a);// adds an element to the end of the array
int pop_back();// removes and returns the last element of the array
int insert(int idx, int val);// inserts the value val at idx. Returns 1 for a
  ↪ successful insertion and -1 if idx does not exists or is invalid. Shift the
  ↪ elements after idx to the right.
int erase(int idx, int val);// erases the value val at idx. Returns 1 for a
  ↪ successful deletion and -1 if idx does not exists or is invalid. Shift the
  ↪ elements after idx to the left.
void size();
int length();// returns the size of the Array
void clear();//clears the contents of the Array
int value(int idx);//returns the value at idx
void assign(int idx, int val);//assigns the value val to the element at index idx
void copy(const Array& Arr);// Copy the passed Array
void copy(const int * arr, int siz);// copy the passed array
void display();// displays the Array
bool isEmpty();// returns true if the Array is empty
Array find(int);// returns an Array containing all the indexes of integer being
  ↪ searched
bool equal(Array);// should return true if both Arrays are same
int sort();// sorts the Array. Returns true if the array is already sorted
void reverse():// reverses the contents of the array
~Array();// destructor...
};
```

Q3: Implementation of Matrix Class Your goal is to implement a generic "Matrix" class using cstring, *i.e.*you will need to write three files (matrix.h, matrix.cpp and matrixMain.cpp). Your implemented class must fully provide the definitions of following class (interface) functions . Please also write down the test code to drive your class implementation. **Please note that we will be running your code against our test code and any segmentation faults or incorrect result will result in loss of marks.**

```cpp
class Matrix{
// think about the private data members...
// the matrix should store real numbers
public:
//include all the necessary checks before performing the operations in the functions
Matrix();// a default constructor
Matrix(int, int);// a parametrized constructor
Matrix(const Matrix &);// copy constructor
void set(int i, int j, float val);//set value at (i,j)
float get(int i, int j)const;//get value at (i,j)
Matrix& assign(const Matrix &);//assigns (copies) a Matrix. Returns the same
Matrix add(const Matrix &);//adds two Matrices and returns the result
Matrix subtract(const Matrix &);//subtracts two Matrices and returns the result
Matrix multiply(const Matrix &);//multiplies two Matrices and returns the result
Matrix multiplyElement(const Matrix &);//Elementwise multiplies two Matrices and
  ↪ returns the result
Matrix add(float );//assigns a constant to every element
Matrix multiply(float);//multiplies every element with a constant
void input(); // takes input in every element of matrix
void display(); // prints every element
~Matrix();
}
```

Q4: Implementation of Molecule Class Your goal is to implement a generic "Molecule" class. Each Molecule Class will consists of collection of Bond class and each Bond will have a collection of Atom Class *i.e.*you will need to write 5 files (molecule.h, molecule.cpp, bond.h, bond.cpp, atom.h, atom.cpp and moleculeMain.cpp). Your implemented class must fully provide the definitions of following class (interface) functions . Please also write down the test code to drive your class implementation. **Please note that we will be running your code against our test code and any segmentation faults or incorrect result will result in loss of marks.**

```
class Molecule{
Bond bondObj[];
// think about other private data members...
public:
// provide definitions of following functions...
Molecule();// default constructor
Molecule(const Molecule &);// copy constructor to initialize the Molecule
  ↪ fromexisting Molecule
AddBond(const Bond&);
PrintFormula() const;
GetBond() const;
~Molecule();// destructor...
};
class Bond{
Atom atomObj[];
// think about other private data members...
public:
// provide definitions of following functions...
Bond();// default constructor
Bond(const Bond &);// copy constructor to initialize the Bond fromexisting Bond
AddBond(const Bond&);
GetBond() const;
~Bond();// destructor...
};
class Atom{
// think about other private data members...
public:
// provide definitions of following functions...
Atom();// default constructor
Atom(const Atom &);// copy constructor to initialize the Atom fromexisting Atom
AddAtom(const Atom&);
GetAtom() const;
~Atom();// destructor...
};
```