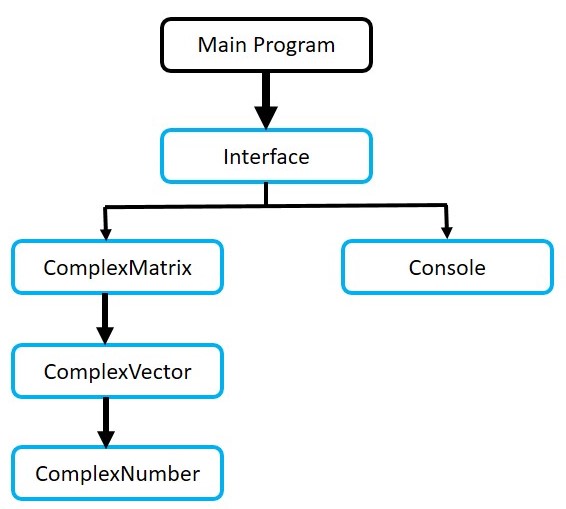
CPSC4110- Quantum Computing

**Project Report**

**Group Members:**

* Brett Reqnier
* Daylend Degrasse
* Mitra Kazemzadeh
* Shah Jabeen Sajid

**Overall Design**

* The basic design of the program is provided on the right side which explains that the main programs calls the interface and interface calls ComplexMatrix and Console class. Further these ComplexMatrix calls ComplexVector that calls the ComplexNumber.
* Interface class implement some main functions such as Toffoli, CNOT and Deutsh’s Algorithm that calls functions in ComplexMatrix and Console class for further implementation.
* Where ComplexMatrix calls ComplexVector and ComplexVector calls ComplexNumber for other functionality.
* The detailed diagram is provided on the last page.

**Running Steps**

* The user can run the program with command “*quantum*”. It runs the main program that calls the interface class.
* It provides the user with three options:

1. CNOT
2. Toffoli
3. Deutsh’s

* At this stage the user can only insert 1, 2 or 3 over any other input the program will output the error.
* Over selecting 1. CNOT the program as for the control bit and the target bit. Here the user can only insert either 1 or 0. Over inserting any other value the program will output the error. The program outputs the right result after the input is inserted.
* Over selecting 2. Toffoli, the program will ask the user to insert first control bit, the second control bit and target bit. Here the user can only insert 1 or 0 for all these options.
* Over selecting 3. Deutsh’s, the prorgam will ask the user for f(0) and f(1) value that can be either 1 or 0. The program tells whether it is a constant case or balanced case further it outputs the qubit.
* After giving the output of a specific function the program again gives the options to select until the user terminated the program.

**Implementation**

The program incorporates fives classes for implementing the CNOT function, Toffoli function and Deutsh’s Algorithm.

1. ComplexNumber

* It consist of two data members i.e real value and imaginary value.
* It consist of some functions that are allocated for a specific complex number passed into it such as add, product, conjugate, real() and imaginary().

2. ComplexVector

* It does not have any data member.
* It consist of some functions that are allocated for a specific complex vector passed into it. It is considered as a complex matrix buth with just a column.
* It consist of functions such as Add, DotProduct, Conjugate, insert.

3. ComplexMatrix

* It has three data members,a vector of complex vectors,a complex number as scalar to the matrix and one boolean to check the dimension of the matrix.
* It has functions to make a complex vector with entered number of rows and columns,overloading operators(=,+,[]), to add and multiply two complex vectors,to check if it is hermitian, identity or unitary, to return the conjugate , Adjoint or transpose of a matrix, a function to return the tensor product of two matrices, It also has some functions to return the 4\*4 CNOT matrix, 2\*2 Hadamard matrix, 8\*8 TOFFOLI matrix.

4. Interface

* This class is to interact with the user, giving the user options to choose from and getting users input and showing the results accordingly.
* It has two data members,*option*:to hold user’s chosen option,and *qubit* which is a 2\*1 complex matrix and would be initialized to either (0 , 1) or (1 , 0) based on users input.
* Additional to constructors it has ComplexMatrix qubit\_input(ComplexMatrix a,int input) function which would set the passed complex matrix to |0> or |1> based on the passed integer.
* ComplexMatrix create\_qubit(int v) would just make a complex vector based on the passed integer.
* ComplexMatrix ControlNOT\_function() would implement the control-Not gate.It would take two inputs as control-bit and the input,as 0 or 1,and calculate their tensor product to have 4\*1 complex matrix as the input and calculate the CNOT of that matrix, and return the result as a 4\*1 complex matrix

5. Console

* This class is used for taking the input from the user and printing the result on the console.
* It has functions like Print, PrintIn and GetInteger that takes in a string and process it for a specific functionality.

