```
//**********************************************************/
// Exercise 05: Sort Data & Simulate Queue by YH in 2017/11
//**********************************************************/
#include "DS1JobList.hpp"                    // type jobType, class JobList
#include "DS1JobQueue.hpp"                   // class JobQueue
#include "DS1AnsList.hpp"                    // class AnsList

void sortData(JobList &);                    // declaration: Mission 1.
void simulateJQ(JobList &);                  // declaration: Mission 2.

//****************************************************************/
// MAIN function is as below
//****************************************************************/
int main(void)
{   int          command = 0;                // user command
    JobList      inputList;                  // job list

    do
    {   cout << endl << "**** Shell Sort & FIFO Queue ***";
        cout << endl << "* 0. Quit                    *";
        cout << endl << "* 1. Sort a file by Shell Sort *";
        cout << endl << "* 2. Simulate one FIFO queue   *";
        cout << endl << "*********************************";
        cout << endl << "Input a command(0, 1, 2): ";
        cin >> command;                      // get the command
        switch (command)
        { case 0: break;
            case 1: sortData(inputList);     // call: Mission 1.
                    break;
            case 2: if (!inputList.isEmpty())// a job list exists
                        simulateJQ(inputList);// call: Mission 2.
                    else    cout << endl << "Please choose command 1 first!" << endl;
                    break;
            default: cout << endl << "Command does not exist!" << endl;
        } // end switch
    } while (command != 0);                  // '0': stop the program
    return 0;
}    // end of main//*******************************************************/
```

```
//*******************************************************/
// Header file for Job List by YH in 2017/11
//*******************************************************/
using namespace std;                                    // standard naming space

#include <iostream>                                     // cout, endl
#include <fstream>                                      // open, is_open
#include <string>                                       // string
#include <vector>                                       // vector
#include <cstdlib>                                      // atoi, system
#include <iomanip>                                      // setw, setprecision
#include <ctime>                                        // clock, CLOCKS_PER_SEC

typedef struct jT
{    int jobID;                                         // job identifier
     int arrival;                                       // arrival time
     int duration;                                      // job duration
     int timeOut;                                       // expire time
}    jobType;

class JobList {
     vector<jobType> aList;                             // list of jobs with four columns
     void    reset();                                   // declaration: initial set up
     //*********************************************************************/
     // The above are private
     //*********************************************************************/
public:
     JobList()   {   this->reset();   }                 // constructor for initialization
     bool    isEmpty()   {   return (aList.size() == 0);   }   // check whether it is empty
     bool    getAll();                                  // declaration: read all from a file
     void    sortByArrival();                           // declaration: sort all by arrival time
     void    putAll();                                  // declaration: write all as a file
     void    showTime();                                // declaration: output time on screen
};   //end JobList
//*********************************************************************/
// class JobList declaration is as the above
//*********************************************************************/
```

```
//*********************************************************************/
// Header file for Job Queue by YH in 2017/11
//*********************************************************************/
class JobQueue {
      jobType          *cA;                                    // circular array
      int        qFront, qBack;                          // head & tail of queue
      //*****************************************************************/
      // The above are private
      //*****************************************************************/
public:
      JobQueue (int maxS)                          // constructor of an empty queue
      {     cA = new jobType [maxS];                          // create the array
            …
      }   //end constructor
      bool isEmpty();                                    // check whether it is empty
      bool isFull();                                     // check whether it is full
      void enQueue(jobType &newItem);                    // append a new element
      void getFront(jobType &oldItem);                   // get the first element
      void deQueue();                                    // drop the first element
      void deQueue(jobType &oldItem)               // get & drop the first element
      {     getFront(oldItem);
            deQueue();
      }     //end deQueue

      ~JobQueue()                                                // destructor
      {     while (!isEmpty())
                  deQueue();
            delete [] cA;
            cA = NULL;
      }     //end destructor
};    //end JobQueue
//*********************************************************************/
// class JobQueue declaration is as the above
//*********************************************************************/

//*********************************************************************/
// Header file for Answer List by YH in 2017/11
//*********************************************************************/
```

```
using namespace std;                                    // standard naming space

#include <iostream>                                                 // cout, endl
#include <fstream>                                                  // open, is_open
#include <string>                                                   // string
#include <vector>                                                   // vector
#include <cstdlib>                                                  // atoi, system
#include <iomanip>                                                  // setw, setprecision

class AnsList {

    typedef struct aT
    {   int jobID;                                                  // abort job identifier
        int abort;                                                  // time to abort
        int delay;                                                  // job waiting time
    }abortType;

    typedef struct dT
    {   int jobID;                                                  // done job identifier
        int depart;                                                 // departure time
        int delay;                                                  // job waiting time
    }doneType;

    vector<abortType>   abortJobs;              // list of aborted jobs with three columns
    vector<doneType>    doneJobs;               // list of finished jobs with three columns
    //*****************************************************************/
    // The above are private
    //*****************************************************************/
public:
    AnsList()   {   abortJobs.clear();   doneJobs.clear();   }      // constructor
    void  showAll();                            // definition: display all on screen
    void  addAbortJob();                        // definition: add one aborted job
    void  addDoneJob();                         // definition: add one finished job
    void  putAll();                             // declaration: write all as a file
};  //end AnsList
//*****************************************************************/
// class AnsList declaration is as the above
//*****************************************************************/
```