Task 1 (C++) Example of class ArrayBub implementing the Bubble Sort.

[bubbleSort.cpp]

```
//bubbleSort.cpp
//demonstrates bubble sort
#include <iostream>
#include <vector>
using namespace std;
//--------------------------------------------------------------
class ArrayBub
{
    private:
        vector<double> v;                   //vector v
        int nElems;                         //number of data items
//--------------------------------------------------------------
        void swap(int one, int two)         //private member function
        {
            double temp = v[one];
            v[one] = v[two];
            v[two] = temp;
        }
//--------------------------------------------------------------
    public:
//--------------------------------------------------------------
        ArrayBub(int max) : nElems(0)       //constructor
        {
            v.resize(max);                  //size the vector
        }
//--------------------------------------------------------------
        void insert(double value)           //put element into array
        {
            v[nElems] = value;              //insert it
            nElems++;                       //increment size
        }
//--------------------------------------------------------------
        void display()                      //displays array contents
        {
            for(int j=0; j<nElems; j++)     //for each element,
                cout << v[j] << " ";        //display it
            cout << endl;
        }
//--------------------------------------------------------------
        void bubbleSort()                   //sorts the array
        {
            int out, in;

            for(out=nElems-1; out>1; out--)   //outer loop (backward)
                for(in=0; in<out; in++)       //inner loop (forward)
                    if( v[in] > v[in+1] )     //out of order?
                        swap(in, in+1);       //swap them
        }  //end bubbleSort()
//--------------------------------------------------------------
};  //end class ArrayBub
////////////////////////////////////////////////////////////////
int main()
{
    int maxSize = 100;                      //array size
    ArrayBub arr(maxSize);                  //create the array

    arr.insert(77);                         //insert 10 items
```

```
    arr.insert(99);
    arr.insert(44);
    arr.insert(55);
    arr.insert(22);
    arr.insert(88);
    arr.insert(11);
    arr.insert(00);
    arr.insert(66);
    arr.insert(33);

    arr.display();                          //display items
    arr.bubbleSort();                       //bubble sort them
    arr.display();                          //display them again
    return 0;
}  //end main()
```

Task 2 (C++) Example of class ArrayIns implementing the Insertion Sort.

[insertSort.cpp]

```
//insertSort.cpp
//demonstrates insertion sort
#include <iostream>
#include <vector>
using namespace std;
//--------------------------------------------------------------
class ArrayIns
{
    private:
        vector<double> v;                   //vector v
        int nElems;                         //number of data items
//--------------------------------------------------------------
    public:
        ArrayIns(int max) : nElems(0)       //constructor
        {
            v.resize(max);                  //size the vector
        }
//--------------------------------------------------------------
        void insert(double value)           //put element into array
        {
            v[nElems] = value;              //insert it
            nElems++;                       //increment size
        }
//--------------------------------------------------------------
        void display()                      //displays array contents
        {
            for(int j=0; j<nElems; j++)     //for each element,
                cout << v[j] << " ";        //display it
            cout << endl;
        }
//--------------------------------------------------------------
        void insertionSort()
        {
            int in, out;
            for(out=1; out<nElems; out++)       //out is dividing line
            {
                double temp = v[out];           //remove marked item
                in = out;                       //start shifts at out
```

```
                while(in>0 && v[in-1] >= temp)  //until one is smaller,
                {
                    v[in] = v[in-1];            //shift item to right
                    --in;                       //go left one position
                }
                v[in] = temp;                   //insert marked item
            }  //end for
        }  //end insertionSort()
//--------------------------------------------------------------
};  //end class ArrayIns
////////////////////////////////////////////////////////////////
int main()
{
    int maxSize = 100;              //array size
    ArrayIns arr(maxSize);          //create array

    arr.insert(77);                 //insert 10 items
    arr.insert(99);
    arr.insert(44);
    arr.insert(55);
    arr.insert(22);
    arr.insert(88);
    arr.insert(11);
    arr.insert(00);
    arr.insert(66);
    arr.insert(33);

    arr.display();                  //display items
    arr.insertionSort();            //insertion-sort them
    arr.display();                  //display them again
    return 0;
}  //end main()
```

Task 3 (C++) Example of sorting objects by using Insertion Sort algorithm.

[objectSort.cpp]

```
//objectSort.cpp
//demonstrates sorting objects (uses insertion sort)
#include <iostream>
#include <string>
#include <vector>
using namespace std;

class Person
{
    private:
        string lastName;
        string firstName;
        int age;

    public:
        Person(string last, string first, int a) :   //constructor
            lastName(last), firstName(first), age(a)
        {  }

        void displayPerson()
```

```
        {
            cout << "   Last name: " << lastName;
            cout << ", First name: " << firstName;
            cout << ", Age: " << age << endl;
        }

        string getLast()                            //get last name
        { return lastName; }
};  //end class Person
//-------------------------------------------------------------
class ArrayInOb
{
    private:
        vector<Person*> v;                  //vect of ptrs to Persons
        int nElems;                         //number of data items

    public:
        ArrayInOb(int max) : nElems(0)      //constructor
        {
            v.resize(max);                  //size the vector
        }

        //put person into array
        void insert(string last, string first, int age)
        {
            v[nElems] = new Person(last, first, age);
            nElems++;                       //increment size
        }

        void display()                      //displays array contents
        {
            for(int j=0; j<nElems; j++){    //for each element,
                cout << v[j]; v[j]->displayPerson();       //display it
            }
        }

        void insertionSort()
        {
            int in, out;
            for(out=1; out<nElems; out++)
            {
                Person* temp = v[out];      //out is dividing line
                in = out;                   //start shifting at out
                                            //until smaller one found,
                while( in>0 && v[in-1]->getLast() > temp->getLast() )
                {
                    v[in] = v[in-1];        //shift item to the right
                    --in;                   //go left one position
                }
                v[in] = temp;               //insert marked item
            }  //end for
        }  //end insertionSort()

};  //end class ArrayInOb
//-------------------------------------------------------------
int main()
{
    int maxSize = 100;              //array size
    ArrayInOb arr(maxSize);         //create array

    arr.insert("Evans", "Patty", 24);
```

```
    arr.insert("Smith", "Doc", 59);
    arr.insert("Smith", "Lorraine", 37);
    arr.insert("Smith", "Paul", 37);
    arr.insert("Yee", "Tom", 43);
    arr.insert("Hashimoto", "Sato", 21);
    arr.insert("Stimson", "Henry", 29);
    arr.insert("Velasquez", "Jose", 72);
    arr.insert ("Vang", "Minh", 22);
    arr.insert("Creswell", "Lucinda", 18);

    cout << "Before sorting:" << endl;
    arr.display();                  //display items

    arr.insertionSort();            //insertion-sort them

    cout << "After sorting:" << endl;
    arr.display();                  //display them again
    return 0;
}  //end main()
```

Based on: Robert Lafore – "Teach Yourself Data Structures And Algorithms In 24 Hours"