# Homework 9

**The goal of this homework is to practice template method pattern. In the sorting algorithm that we implement using the abstract factory pattern, the source code of which is in http://www.cse.msu.edu/~alexliu/courses/335Spring2019/lectures/ExampleSourceCode_C++AbstractFactorySorting.zip, recall that we implemented two algorithms sortDecreasing and sortIncreasing in the algorithm class BubbleSort. In this homework, you are asked to use the template method pattern to refactor (i.e., redesign) this algorithm class. You also need to draw the UML diagram. You need to design classes so that the following main function will output the following. Your program must use this main function.**

```
********** Before Sorting Integers Decreasing
5;
4;
6;
10;
********** After Sorting Integers Decreasing
10;
6;
5;
4;
********** After Sorting Integers Increasing
4;
5;
6;
10;

RUN SUCCESSFUL (total time: 90ms)
```

```
int main(int argc, char** argv) {
   IntegerVectorSortable ivs;
   ivs.insertInteger(5);
   ivs.insertInteger(4);
   ivs.insertInteger(6);
   ivs.insertInteger(10);

   cout<<"********** Before Sorting Integers Decreasing"<<endl;
   ivs.print();
   cout<<"********** After Sorting Integers Decreasing"<<endl;
   BubbleSortDecreasing bsd;
   bsd.sort(&ivs);
   ivs.print();
   cout<<"********** After Sorting Integers Increasing"<<endl;
   BubbleSortIncreasing bsi;
   bsi.sort(&ivs);
   ivs.print();

   return 0;
}
```

**Due: February 13th, 11:59PM, 2019.**

**Turn in one file via handin: the zip file of (1) your whole NetBean directory and (2) your UML digram file named uml.pdf. The name of your zip file should be: LastName_FirstName.zip. For example, if your name is John Smith, you should turn in one files: Smith_John.zip.**