

## Homework 8

The goal of this homework is to practice the abstract factory pattern. In the videoed lecture, I have explained how you should use the abstract factory pattern to implement a binary search algorithm. In this homework, you are asked to implement the binary search algorithm using the abstract factory pattern and draw your class diagrams. Your binary search algorithm should work with an abstract interface class called **SearchableVector**. You need to design classes so that the following main program will compile and allow you to input a number to search over an array of sorted data 1, 2, 3, 4, 5, 6, 7, 8, and 9. **Your program must use this main function.**

```
int main(int argc, char** argv) {
    IntegerVectorSearchable ivs;
    ivs.insertInteger(1);
    ivs.insertInteger(2);
    ivs.insertInteger(3);
    ivs.insertInteger(4);
    ivs.insertInteger(5);
    ivs.insertInteger(6);
    ivs.insertInteger(7);
    ivs.insertInteger(8);
    ivs.insertInteger(9);

    BinarySearch bs;

    cout<<"All integers are: "<<endl;
    ivs.print();
    int query = 1;
    while(query!=0){
        cout<<"Please input the number that you want to search: ";
        cin>>query;
        ivs.setQuery(query);
        int searchResult=bs.search(&ivs);
        cout<<endl;
        if(searchResult==-1) cout<<"There is no match!"<<endl;
        else cout<<"Find match at the "<<searchResult<<"th element!"<<endl;
    }
    return 0;
}
```

**Initially, your program prints the following:**

```
All integers are:
1; 2; 3; 4; 5; 6; 7; 8; 9;
Please input the number that you want to search:
```

**After you type in a number, say 1, press return twice, it will print the following and wait for you to type in another number to search.**

```
Find match at the 0th element!
Please input the number that you want to search: 
```

**Finally, if you input 0, then it will print “There is no match!” and terminates.**

**Due: February 9<sup>th</sup>, 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.**