Examination 4

CSC140 Foundations of Computer Science

28 February 2015

Writing in English

1. Read one of these influential articles. Write a summary and response. Write about one page.

   If you prefer, write shorter responses to two articles.

   You may write in the first person. Make your response personal. If you can, make connections between what you read and your own experiences, ambitions, intuition, and beliefs.

   - *Software Engineering Principles*, by Steve McConnell
   - *No Silver Bullet*, by Fred P. Brooks, Jr.
   - *You and Your Research*, by Richard Hamming
   - *How to Became a Hacker*, by Eric Raymond
   - *Plenty of Room at the Bottom*, by Richard Feynman
   - *As We May Think*, by Vannevar Bush
   - *The GNU Manifesto*, by Richard Stallman
   - *Ten Lessons I Wish I Had Been Taught*, by Gino-Carlo Rota
   - *Is Computer Science Science?*, by Peter Denning
   - *Computing is a Natural Science*, by Peter Denning
   - *The Computer Scientist as Toolsmith II*, by Frederick P. Brooks, Jr.
   - *The Humble Programmer*, by Edsger Dijkstra

2. Write two paragraphs about each of three people that you select from the following list. Learn about these people by reading on the Web. Look for biographical details, anecdotes, or quotations that give some sense of personality, motives, historical perspective, or the excitement of contributing to a still young field.

   - Hal Abelson
   - Marc Andreesen
   - Isaac Asimov

- John Atanasoff
- Charles Babbage
- John Backus
- Steve Balmer
- Gordon Bell
- Fred Brooks, Jr.
- Vannevar Bush
- Vint Cerf
- Edgar F. (Ted) Codd
- Stephen Cook
- Seymour Cray
- Peter J. Denning
- Edsger Dijkstra
- John Doerr
- J. Presper Eckert
- Richard Feynman
- Carly Fiorina
- Bill Gates
- Adele Goldberg
- Herman Goldstine
- Shafi Goldwasser
- James Gosling
- Andy Grove
- Richard Hamming
- John Hennessy
- Danny Hillis
- C.A.R. (Tony) Hoare
- Herman Hollerith
- Grace Murray Hopper
- Steve Jobs
- Bob Kahn
- Richard Karp
- Gary Kasparov
- John Kemeny
- Vinod Khosla

- Jack Kilby
- Donald E. Knuth
- Ray Kurzweil
- John Lasseter
- Gottfried Wilhelm Leibniz
- J.C.R. Licklider
- Barbara Liskov
- Ada Lovelace
- John Mauchly
- Marissa Mayer
- John McCarthy
- Marvin Minsky
- Gordon Moore
- Peter Naur
- Robert Noyce
- Ken Olsen
- David Patterson
- Radia Perlman
- Eric Raymond
- Ginni Rometty
- Ken Sakamura
- Claude Shannon
- William Shockley
- Balaji Srinivasan
- Richard Stallman
- Cliff Stoll
- Bjarne Stroustrup
- Ivan Sutherland
- Andy Tanenbaum
- Linus Torvalds
- Alan Turing
- John von Neumann
- Thomas Watson, Jr.
- Meg Whitman
- Norbert Wiener

- Maurice Wilkes
- Jeanette Wing
- Niklaus Wirth
- Steve Wozniak
- Konrad Zuse

Writing in Java and English

Create a new project on the computer. Enter this code. Add comments that describe the purpose of each method. Use Javadoc to generate a Web site that presents the code and your commentary.

```
package inorder;

public class InOrder {

    public static int[] makeData(int n) {
        int[] data = new int[n];
        for (int i = 0; i < data.length; i++) {
            data[i] = 10 + (int) (90 * Math.random());
        }
        return data;
    } // makeData( int )

    public static void printData(int[] data) {
        for (int i = 0; i < data.length; i++) {
            System.out.print(data[i] + " ");
        } // for
        System.out.println();
    } // printData( int [] )

    public static int positionOfMinimum(int[] data, int i) {
        int bestGuessSoFar = i;
        for (int j = i; j < data.length; j++) {
            if (data[j] < data[bestGuessSoFar]) {
                bestGuessSoFar = j;
            } // if
        } // for
        return bestGuessSoFar;
    } // positionOfMinimum( int [], int )

    public static void swap(int[] data, int i, int j) {
        int temp = data[i];
        data[i] = data[j];
```

```
    data[j] = temp;
} // swap( int [], int, int )

public static void selectionSort(int[] data) {
    for (int i = 0; i < data.length; i++) {
        int j = positionOfMinimum(data, i);
        swap(data, i, j);
    } // for
} // selectionSort( int [] )

public static int insertionPoint(int[] data, int i) {
    int j = i;
    while (j > 0 && data[j - 1] > data[i]) {
        j--;
    } // while
    return j;
} // insertionPoint( int [], int )

public static void insert(int[] data, int i, int j) {
    int temp = data[i];
    for (int k = i; k > j; k--) {
        data[k] = data[k - 1];
    } // for
    data[j] = temp;
} // insert( int [], int, int )

public static void insertionSort(int[] data) {
    for (int i = 0; i < data.length; i++) {
        int j = insertionPoint(data, i);
        insert(data, i, j);
    } // for
} // insertionSort( int [] )

public static int[] merge(int[] a, int[] b) {
    int[] result = new int[a.length + b.length];

    int i = 0;
    int j = 0;
    int k = 0;
    while (i < a.length && j < b.length) {
        if (a[i] < b[j]) {
            result[k] = a[i];
            i++;
        } // if
        else {
            result[k] = b[j];
```

```
            j++;
        } // else
        k++;
    } // while

    while (i < a.length) {
        result[k] = a[i];
        k++;
        i++;
    } // while

    while (j < b.length) {
        result[k] = b[j];
        k++;
        j++;
    } // while

    return result;
} // merge( int [], int [] )

public static int[] sublist(int[] data, int start, int end) {
    int[] result = new int[end - start + 1];
    for (int i = start; i <= end; i++) {
        result[i - start] = data[i];
    } // for
    return result;
} // sublist( int [], int, int )

public static int[] mergeSort(int[] data) {
    if (data.length == 1) {
        return data;
    } // if
    else {
        int start = 0;
        int end = data.length / 2;
        int[] a = sublist(data, start, end - 1);
        int[] b = sublist(data, end, data.length - 1);
        return merge(mergeSort(a), mergeSort(b));
    } // else
} // mergeSort( int [] )

public static void main(String[] args) {
    System.out.println( "Selection sort.");
    int[] data = makeData(8);
    printData(data);
    selectionSort( data );
```

```
        printData(data);

        System.out.println();

        System.out.println( "Insertion sort." );
        data = makeData(8);
        printData(data);
        insertionSort( data);
        printData( data );

        System.out.println();

        System.out.println( "Merge sort." );
        data = makeData(8);
        printData(data);
        int [] result = mergeSort( data );
        printData( result );
    } // main( String [] )

} // InOrder
```