Insertion sort

```java
public void insertionSort(int[] A){
    for (int j = 1; j < A.length; j++) {
        int key = A[j];
        //Insert A[j] into the sorted sequence A[1..j-1]
        int i = j - 1;
        while (i >= 0 && A[i] > key) {
            A[i + 1] = A[i];
            i--;
            A[i + 1] = key;
        }
    }
}
```

```java
        object.insertionSort(array);
        for (int i = 0; i < array.length; i++) {
            System.out.print(array[i]);
        }

    }
}
```

Javadoc Declaration Console ×

<terminated> Chapter2Programming [Java Application] C:\Users\zaybd\.p2\p

```
122344456
```

2.1-2

```
22    public void reverseInsertionSort (int[] A) {
23        for (int j = 1; j < A.length; j++) {
24            int key = A[j];
25            //Insert A[j] into the sorted sequence A[1..j-1]
26            int i = j - 1;
27            while (i >= 0 && A[i] < key) {
28                A[i + 1] = A[i];
29                i--;
30                A[i + 1] = key;
31            }
32        }
33    }
```

```
146         object.reverseInsertionSort(array);
147         for (int i = 0; i < array.length; i++) {
148             System.out.print(array[i]);
149         }
150
151     }
152
153 }
```

Javadoc | Declaration | Console ×

<terminated> Chapter2Programming [Java Application] C:\Users\zaybd\.p2\po

654443221

2.1-4

```java
public boolean[] binaryAddition (boolean[] a, boolean[] b) throws Exception {

    if (a.length != b.length) {
        throw new Exception("The arrays are different sizes");
    }

    boolean[] result = new boolean[a.length + 1];

    for (int i = 0; i < a.length; i++) {
        //If they are both 0s or both 1s, result will start false
        if (a[i] == b[i]) {
            result[i+1] = false;
            if (a[i] == true) {
                int c = i;
                while (result[c] == true && c > 0) {
                    result[c] = false;
                    c--;
                }
                result[c] = true;
            }
        }
        else {
            result[i+1] = true;
        }
    }
    return result;
}
```

```java
boolean[] binaryOne = {true, true, false, false, true};
boolean[] binaryTwo = {false, true, false, true, true};

boolean[] result = null;

try {

    result = object.binaryAddition(binaryOne, binaryTwo);
} catch (Exception e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}

for (int i = 0; i < result.length; i++) {
    System.out.println(result[i]);
}

}
```

Javadoc Declaration Console

<terminated> Chapter2Programming [Java Application] C:\Users\zaybd\.p2\pool\plugins\org.eclipse.ju

```
true
false
false
true
false
false
```

Merge and MergeSort

```java
public void merge (int[] A, int p, int q, int r) {
    int nOne = q - p + 1;
    int nTwo = r - q;
    int[] L = new int[nOne+2];
    int[] R = new int[nTwo+2];
    for (int i = 0; i <= nOne; i++) {
        L[i] = A[p+i];
    }
    for (int j = 0; j <= nTwo; j++) {
        R[j] = A[q+j];
    }
    L[nOne] = Integer.MAX_VALUE;
    R[nTwo] = Integer.MAX_VALUE;
    int i = 0;
    int j = 0;
    int k = p;

    while (k < r)   {
        if (L[i] <= R[j]) {
            A[k] = L[i];
            i++;
        }
        else {
            A[k] = R[j];
            j++;
        }
        k++;
    }
}
```

```java
public void mergeSort (int[] A, int p, int r) {
    if (p < r) {
        int q = ((p+r)/2);
        mergeSort (A, p, q);
        mergeSort (A, q+1, r);
        merge (A, p, q, r);
    }
}
```

2.3-5

```
138     System.out.println("\n");
139         object.mergeSort(array, 0, array.length-1);
140         for (int i = 0; i < array.length; i++) {
141             System.out.print(array[i]+", ");
142         }
143
144
145     /*
146         object.reverseInsertionSort(array);
```

Javadoc Declaration Console ×

```
<terminated> Chapter2Programming [Java Application] C:\Users\zaybd\.p2\poo
3, 1, 4, 5, 2, 4, 9, 6, 4,

1, 2, 3, 4, 4, 2, 4, 9, 4,
```

```
103     public int binarySearch (int[] A, int q, int low, int high) {
104         if (high >= low) {
105                 int mid = low + (high - low) / 2;
106                 if (A[mid] == q) {
107                     return mid;
108                 }
109                 if (A[mid] > q) {
110                     return binarySearch(A, q, low, mid-1);
111                 }
112                 return binarySearch(A, q, mid+1, high);
113             }
114         return -1;
115           }
116
```

```
167     int[] array = {3, 1, 4, 5, 2, 4, 9, 6, 4};
168     System.out.print(object.binarySearch(array, 9, 0, array.length-1));
169
170
171
172
173     }
174
```

Javadoc Declaration Console ×

```
<terminated> Chapter2Programming [Java Application] C:\Users\zaybd\.p2\pool\plugins\org.eclipse.justj.o
6
```

2.2

```
117     public void bubbleSort (int[] A) {
118         for (int i = 0; i < A.length-1; i++) {
119             for (int j = A.length-1; j > i; j--) {
120                 if (A[j] < A[j-1]) {
121                     int temp = A[j];
122                     A[j] = A[j-1];
123                     A[j-1] = temp;
124                 }
125             }
126         }
127     }
```

```
166         int[] array = {3, 1, 4, 5, 2, 4, 9, 6, 4};
167         object.bubbleSort(array);
168           for (int i = 0; i < array.length; i++) {
169            System.out.print(array[i]+", ");
170         }
171
172     }
173
174 }
```

Javadoc | Declaration | Console ×

<terminated> Chapter2Programming [Java Application] C:\Users\zaybd\.p2

```
1, 2, 3, 4, 4, 4, 5, 6, 9,
```