UMBER 2

```
ublic String acronymFor(List<String> list) {
       String res = “”;
       for (String s : list){
              res += s.charAt(0);
       }
       return res;
```

UMBER 3

tatement	Output
ar1.method1();	Scoop 1
ar2.method1();	Spoon 1/Cone 1
ar3.method1();	Cone 1
ar4.method1();	compiler error
ar5.method1();	Scoop 1
ar6.method1();	compiler error
ar1.method2();	Cone 2/Scoop 1
ar3.method2();	Cone 2/Cone 1
ar6.method2();	compiler error
(Spoon)var1).method3();	compiler error
(Cup)var6).method3();	Cup 3
(Scoop)var4).method1();	runtime error
(Cone)var6).method2();	Cone 2/Cone 1
(Cone)var4).method1();	Spoon 1/Cone 1
(Scoop)var6).method3();	runtime error

UMBER 4

```
ublic void printLevel(int n){
       if n < 1
              IAG
       printLevel(overallRoot, n - 1);


rivate void printLevel(IntTreeNode curr, int n) {
       if (curr != null) {
              if (n == 0) {
                     System.out.println(curr.data);
              } else {
                     printLevel(curr.right, n-1);
                     printLevel(curr.left, n-1);
              }
       }
```

UMBER 5

```
ublic static Map<String, List<Double>> studentGrades(Map<String, Map<String, Double>> classes) {
     Map<String, List<Double>> result = new TreeMap<String,List<Double>>();
     for (String teacher : classes.keySet()) {
           Map<String, Double> grades = classes.get(teacher);
           for (String student : grades.ketSet()) {
                 if (!result.containsKey(student)) {
                       result.put(student, new ArrayList<Double>()) ;
                 }
                 result.get(student).add(grades.get(student));
           }
     }
     return result;
```

UMBER 6

```
ublic boolean sameDashes(String one, String two) {
     if (one.length() != two.length()) {
           throw new IllegalArgumentException();
     }
     if ((one.charAt(0) == '-' || two.charAt(0) == '-') && two.charAt(0) != one.charAt(0)){
           return false;
     return sameDashes(one.substring(1), two.substring(1));
```

UMBER 7

```
ublic boolean bubble() {
       boolean swap = false;
       if (front != null && front.next != null) {
           if (front.data > front.next.data) {
               swap = true;
               ListNode temp = front;
               front = front.next;
               temp.next = front.next;
               front.next = temp;
           }
           ListNode current = front;
           while (current.next != null && current.next.next != null) {
               if (current.next.data > current.next.next.data) {
                   swap = true;
                   ListNode temp = current.next.next;
                   current.next.next = temp.next;
                   temp.next = current.next;
                   current.next = temp;
               }
               current = current.next;
           }
       }
       return swap;
   }
```

UMBER 8

(1)
(n)

(log(n))
(n) (shift all elements to the right, O(1) also accepted for replacement)

ndices examined: 24 62 41 33
alue returned: -10

```
/Returns negative value if this TA is less than Other
/Returns positive value if this TA greater than Other
/Returns 0 if TAs the same value
/Accepts TA to be compared to
ublic int compareTo(TeachingAssistant other){
```

UMBER 9

```
ublic IntTree(int n) {
       if (n < 0) {
          throw new IllegalArgumentException();
       }
       overallRoot = makeTree(1, n);


rivate IntTreeNode makeTree(int low, int high) {
    if (low > high) {
       return null;
    } else {
       int mid = (low + high - 1) / 2;
       return new IntTreeNode(high, makeTree(low, mid),
                                       makeTree(mid + 1, high - 1));             }
```