Fall 2013 **Exam #3 Key** COSC 117

1 Program Traces (15 Points Each)

1. For the following questions you will be using this class.

```
1  public class Thing {
2      private int n = 0;
3      private double d = 0;
4      private String s = "";
5
6      public Thing(int a, double b, String c){
7          n = a;
8          d = b;
9          s = c;
10     }
11
12     public void print(){
13         System.out.println(n + " - " + d +
14             " - " + s);
15     }
16
17     public void printNums(){
18         System.out.println(n + " - " + d);
19     }
20
21     public void doOne(int t){
22         n += t;
23     }
24
25     public void doOne(double t){
26         d += t;
27     }
28
29     public void doOne(){
30         n += n;
31     }
32
33     public void up(){
34         n++;
35     }
36
37     public void down(){
38         n--;
39     }
40
41     public boolean isZero(){
42         return n == 0;
43     }
44
45     public boolean isLess(int j){
46         return n < j;
47     }
48
49     public String getString(){
50         return s;
51     }
52
53     public int getInt(){
54         return n;
55     }
56
57     public double getDouble(){
58         return d;
59     }
60
61     public void change(int i){
62         char c = s.charAt(i);
63         String str = "" + c;
64
65         if (c >= 'A' && c <= 'Z')
66             str = str.toLowerCase();
67         else
68             str = str.toUpperCase();
69
70         String bstr = s.substring(0, i);
71         String astr = s.substring(i+1);
72         s = bstr + str + astr;
73     }
74
75     public void UP(){
76         int num = 0;
77         for (int i = 0; i < s.length(); i++){
78             if (s.charAt(i) >= 'A' &&
79                         s.charAt(i) <= 'Z')
80                 num++;
81         }
82
83         change(num);
84
85         if (num % 2 == 0){
86             d = d + num;
87         } else {
88             d = d + 2*num;
89         }
90     }
91 }
```

(a) Write the output of the following code fragment.

```
1  Thing t = new Thing(2, 5.0, "JackAndJill");
2  for (int i = 0; i < t.getString().length(); i
       +=3){
3      t.change(i);
4      t.print();
5  }
```

Program Output

```
2 - 5.0 - jackAndJill
2 - 5.0 - jacKAndJill
2 - 5.0 - jacKAnDJill
2 - 5.0 - jacKAnDJiLl
```

(b) Write the output of the following code fragment.

```
1  Thing r = new Thing(4, 3.0, "JackAndJill");
2  for ( ; !r.isZero(); r.down()){
3      r.UP();
4      r.print();
5  }
```

Program Output

```
4 - 9.0 - JacKAndJill
3 - 13.0 - JacKandJill
2 - 19.0 - JackandJill
1 - 21.0 - JaCkandJill
```

(c) Write the output of the following code fragment.

```
1  Thing s = new Thing(-5, 4.0, "John Smith");
2  s.doOne();
3  s.printNums();
4  for (int i = 0; i <= 3; i++)
5      s.up();
6  s.printNums();
7  s.doOne(17);
8  s.printNums();
9  s.doOne(1.0*s.getInt());
10 s.printNums();
11 while (!s.isLess(0)){
12     s.doOne(-2);
13     s.down();
14     s.doOne(1.1);
15     s.printNums();
16 }
```

Program Output

```
-10 - 4.0
-6 - 4.0
11 - 4.0
11 - 15.0
8 - 16.1
5 - 17.2
2 - 18.3
-1 - 19.4
```

2. Write the output of the following program.

```
1  public class Exam3Trace2 {
2
3      public static void PrintArray(int A[]) {
4          for (int i = 0; i < A.length; i++) {
5              System.out.print(A[i] + "  ");
6          }
7          System.out.println();
8      }
9
10     public static void main(String[] args) {
11         int A[] = new int [5];
12         int B[] = new int [5];
13
14         for (int i = 0; i < 5; i++){
15             A[i] = i+1;
16         }
17         PrintArray(A);
18
19         for (int j = 0; j < 2; j++){
20             for (int i = 0; i < 4; i++){
21                 A[i] += A[i+1];
22             }
23             PrintArray(A);
24         }
25
26         System.out.println("--------------");
27
28         for (int i = 0; i < 5; i++){
29             A[i] = 5-i;
30             B[i] = i+1;
31         }
32
33         PrintArray(A);
34         PrintArray(B);
35
36         A[2] = B[4];
37         A[0] = B[2];
38         B[3] = A[2];
39
40         PrintArray(A);
41         PrintArray(B);
42
43         System.out.println("--------------");
44
45         A = B;
46         PrintArray(A);
47         PrintArray(B);
48
49         B[2] = 7;
50
51         PrintArray(A);
52         PrintArray(B);
53
54         System.out.println("--------------");
55
56         for (int i = 0; i < 5; i++){
57             A[i] = 5-i;
58             B[i] = i+1;
59         }
60
61         PrintArray(A);
62         PrintArray(B);
63
64         System.out.println("--------------");
65     }
66 }
```

Program Output

```
1  2  3  4  5
3  5  7  9  5
8  12  16  14  5
--------------
5  4  3  2  1
1  2  3  4  5
3  4  5  2  1
1  2  3  5  5
--------------
1  2  3  5  5
1  2  3  5  5
1  2  7  5  5
1  2  7  5  5
--------------
1  2  3  4  5
1  2  3  4  5
--------------
```

3. Write the output of the following program.

```
1  public class Exam3Trace3 {
2
3      public static void Print2DArray(int A[][])
4      {
5          int dim1 = A.length;
6          int dim2 = A[0].length;
7          for(int i = 0; i < dim1; i++){
8              for(int j = 0; j < dim2; j++){
9                  System.out.printf(A[i][j]);
10             }
11             System.out.println();
12         }
13     }
14
15     public static int[][] Find(int A[][]) {
16         int[][] thing = new int[2][3];
17
18         int dim1 = A.length;
19         int dim2 = A[0].length;
20         thing[0][2] = A[0][0];
21         thing[1][2] = A[0][0];
22
23         for(int i = 0; i < dim1; i++){
24             for(int j = 0; j < dim2; j++){
25                 if (A[i][j] > thing[0][2]){
26                     thing[0][2] = A[i][j];
27                     thing[0][0] = i;
28                     thing[0][1] = j;
29                 }
30                 if (A[i][j] < thing[1][2]){
31                     thing[1][2] = A[i][j];
32                     thing[1][0] = i;
33                     thing[1][1] = j;
34                 }
35             }
36         }
37         return thing;
38     }
39
40     public static void main(String[] args) {
41         int arr[][] = new int[3][4];
42         int arr2[][] = new int[2][2];
43
44         for (int i = 0; i < 3; i++)
45             for (int j = 0; j < 4; j++)
46                 arr[i][j] = 2*i + j;
47
48         Print2DArray(arr);
49         arr2 = Find(arr);
50         Print2DArray(arr2);
51
52         System.out.println("-----------------");
53
54         int num = arr[arr2[0][0]][arr2[0][1]];
55         arr[arr2[0][0]][arr2[0][1]] = num/2;
56
57         num = arr[arr2[1][0]][arr2[1][1]];
58         arr[arr2[1][0]][arr2[1][1]] = num+2;
59
60         Print2DArray(arr);
61         arr2 = Find(arr);
62         Print2DArray(arr2);
63
64         System.out.println("-----------------");
65
66         int c = 1;
67         for (int i = 0; i < 3; i++)
68             for (int j = 0; j < 4; j++)
69                 arr[i][j] = c++;
70
71         Print2DArray(arr);
72
73         System.out.println("-----------------");
74
75         for (int t = 1; t < 4; t++)
76             arr[2*t % 3][3*t % 4] = arr[(t+2) %
                    3][(t+1) % 4];
77
78         Print2DArray(arr);
79         arr2 = Find(arr);
80         Print2DArray(arr2);
81
82         System.out.println("-----------------");
83     }
84 }
```

Program Output

```
   0   1   2   3
   2   3   4   5
   4   5   6   7
   2   3   7
   0   0   0
----------------
   2   1   2   3
   2   3   4   5
   4   5   6   3
   2   2   6
   0   1   1
----------------
   1   2   3   4
   5   6   7   8
   9  10  11  12
----------------
   1   9   3   4
   5   6   8   8
   9  10  11   3
   2   2  11
   0   0   1
----------------
```

2 Coding

1. (*15 Points*) Write a method called `LinearSearch` that takes two parameters, a one-dimensional array `A` of integers and an integer target value called `N`. The method should return the first position of the target value stored in the array and if the target value is not in the array it should return -1.

```
static int linearSearch(int[] A, int N) {
    for (int index = 0; index < A.length; index++) {
        if (A[index] == N)
            return index;
    }
    return -1;
}
```

2. (*15 Points*) Write either the Bubble Sort, Insertion Sort or Selection Sort (just do one of them) in a method that takes in one parameter only, a one-dimensional integer array named A.

```
public static void BubbleSort(int A[]) {
    for (int i = A.length - 1; i > 0; i--) {
        for (int j = 0; j < i; j++) {
            if (A[j] > A[j + 1]) {
                int temp = A[j];
                A[j] = A[j + 1];
                A[j + 1] = temp;
            }
        }
    }
}

public static void insertionSort(int[] A) {
    for (int itemsSorted = 1; itemsSorted < A.length; itemsSorted++) {
        int temp = A[itemsSorted];
        int loc = itemsSorted - 1;
        while (loc >= 0 && A[loc] > temp) {
            A[loc + 1] = A[loc];
            loc = loc - 1;
        }
        A[loc + 1] = temp;
    }
}

public static void selectionSort(int[] A) {
    for (int lastPlace = A.length - 1; lastPlace > 0; lastPlace--) {
        int maxLoc = 0;
        for (int j = 1; j <= lastPlace; j++)
            if (A[j] > A[maxLoc])
                maxLoc = j;

        int temp = A[maxLoc];
        A[maxLoc] = A[lastPlace];
        A[lastPlace] = temp;
    }
}
```

3. (*15 Points*) Write two array methods, `ReverseArray` and `isPalArray` for the following program. They both take only a one dimensional array of integers as their parameter. `ReverseArray` returns an array of the same size that is the reverse of the input array. `isPalArray` returns a boolean, true if the array is the same if read either forwards or backwards and false if not. The input array is not to be altered in either method.

```
1  public class ArrayReverse {
2
3      public static void PrintArray(int A[]) {
4          for (int i = 0; i < A.length; i++) {
5              System.out.print(A[i] + "  ");
6          }
7          System.out.println();
8      }
9
10     public static int[] CopyArray(int A[]) {
11         int[] B = new int[A.length];
12
13         for (int i = 0; i < A.length; i++)
14             B[i] = A[i];
15
16         return B;
17     }
18
19     public static int[] ReverseArray(int [] A){
20 <<<  Insert ReverseArray Code >>>
21     }
22
23     public static boolean isPalArray(int [] A){
24 <<<  Insert isPalindromeArray Code >>>
25     }
26
27     public static void main(String[] args) {
28         int n = 10;
29         int A[] = new int [n];
30         int B[] = new int [n];
31
32         for (int i = 0; i < n; i++){
33             A[i] = (int)(Math.random()*200-100);
34         }
35
36         B = ReverseArray(A);
37         PrintArray(A);
38         PrintArray(B);
39
40         for (int i = 0; i < n; i++){
41             A[i] += B[i];
42         }
43
44         PrintArray(A);
45         System.out.println(isPalArray(A));
46         System.out.println(isPalArray(B));
47     }
48 }
```

ReverseArray Code

```
int n = A.length;
int[] B = new int[A.length];
for (int i = 0; i < n; i++){
    B[i] = A[n-1-i];
}
return B;
```

isPalArray Code

```
int n = A.length;
for (int i = 0; i < n/2; i++){
    if (A[i] != A[n-1-i])
        return false;
}
return true;
```

Program Run

```
-5  -76  50  -25  -17  93  -36  51  -73  86
86  -73  51  -36  93  -17  -25  50  -76  -5
81  -149  101  -61  76  76  -61  101  -149  81
true
false
```

4. (*20 Points*) Write a `Student` class that stores a student's name in the variable `Name` and an array of 5 doubles called `Grades`. The class should have four methods, each of them is discussed below. There is also a sample main program and its output that uses this class.

(a) `public Student(String n)`

The constructor should store the input string in the Name variable and set all of the grades to -1.

(b) `public void printStudentInfo()`

This method should print out the student's name, list of current grades and their current class average. As shown in the program run to the right.

(c) `public void addGrade(double score)`

This method should add the new grade (score) into the grades array at the first position of a -1. So the grades array will have a list of current grades and -1 after that. So if the grades 87, 73 and 82 have been entered, as in the middle output of the example to the right, the Grades array will look like,

87	73	82	-1	-1

then if we add in the 91 it would be

87	73	82	91	-1

If there are already 5 grades in the array another call to `addGrade` will not put anything into the array.

(d) `public double avg()`

This method returns the student's current average, which is calculated using only the entries of the grades array that are greater than or equal to 0. If no grades have been input this method should return a 0 for the average.

Main Program

```
public class StudentMain {

    public static void main(String[] args) {
        Student stu = new Student("Don
            Spickler");
        stu.printStudentInfo();
        stu.addGrade(87);
        stu.addGrade(73);
        stu.addGrade(82);
        stu.printStudentInfo();
        stu.addGrade(91);
        stu.addGrade(79);
        stu.printStudentInfo();
    }
}
```

Program Run

```
Don Spickler
Grades:
Average: 0.0

Don Spickler
Grades: 87.0  73.0  82.0
Average: 80.66666666666667

Don Spickler
Grades: 87.0  73.0  82.0  91.0  79.0
Average: 82.4
```

Code for the Student Class

```
1  public class Student {
2      private String Name = "";
3      private double Grades[] = new double[5];
4
5      public Student(String n){
6          Name = n;
7          for (int i = 0; i < 5; i++)
8              Grades[i] = -1;
9      }
10
11     public void printStudentInfo(){
12         System.out.println(Name);
13         System.out.print("Grades: ");
14         for (int i = 0; i < 5; i++)
15             if (Grades[i] >= 0)
16                 System.out.print(Grades[i]+ "  ");
17         System.out.println();
18         System.out.println("Average: " + avg());
19         System.out.println();
20     }
21
22     public void addGrade(double score){
23         int pos = 0;
24         while (pos < 5 && Grades[pos] >= 0)
25             pos++;
26
27         if (pos < 5)
28             Grades[pos] = score;
29     }
30
31     public double avg(){
32         int count = 0;
33         double sum = 0;
34         for (int i = 0; i < 5; i++){
35             if (Grades[i] > 0){
36                 sum += Grades[i];
37                 count++;
38             }
39         }
40         if (count == 0)
41             return 0;
42         else
43             return sum/count;
44     }
45 }
```