COSC 117 **Exam 3 Key** Fall 2012

Part 1: Definitions & Short Answer (3 Points Each)

1. A method in a class that has no return type and the same name as the class is called what? How is this type of method called from the main function?

   It is a constructor and is called using the new statement, for example,
   ```
   MyStuff m = new MyStuff();
   ```

2. What is the difference between a private class member and a public class member? All member variables should be which type?

   Public members can be accessed from outside the class and private members can only be accessed from inside the class. All data members should be private.

3. Write a declaration that will create a one-dimensional integer array that will hold 20 items. Then write a segment of code that will put the numbers 3, 5, 1, -2, and 4 in the first 5 entries of the array and the numbers 100, 200, 300, … in the remainder of the array, using a loop.

   ```
   int[] D = new int[20];
   D[0] = 3;
   D[1] = 5;
   D[2] = 1;
   D[3] = -2;
   D[4] = 4;
   for (int i = 0; i < 15; i++)
         D[i + 5] = 100 * i;
   ```

4. Write a declaration that will create a two-dimensional array of doubles that has 10 rows and 23 columns.

   ```
   double[][] A = new double[10][23];
   ```

5. Write a function header that will bring in as a parameter a two-dimensional array of doubles and return a one-dimensional array of integers.

   ```
   public static int[] doSomething(double[][] A)
   ```

Part 2: Program Traces (15 Points Each)

1. For each of the program inputs below write the output of the program.

```
import java.util.Scanner;

public class Exam3Trace1 {

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

    public static void main(String[] args) {
       Scanner keyboard = new Scanner(System.in);
       System.out.print("Input n: ");
       int n = keyboard.nextInt();

       int[] A = new int[n];

       System.out.print("Input elements: ");
       for (int i = 0; i < n; i++)
          A[i] = keyboard.nextInt();

       PrintArray(A);

       int x = 2;
       for (int i = 0; i < n; i++){
          int y = x*x % n;
          int temp = A[x];
          A[x] = A[y];
          A[y] = temp;
          x = y;
       }

       PrintArray(A);
    }
}
```

(a)

```
Input n: 4
Input elements: 1 2 3 4
1 2 3 4
3 2 1 4
```

(b)

```
Input n: 5
Input elements: 1 2 3 4 5
1 2 3 4 5
1 3 5 4 2
```

(c)

```
Input n: 6
Input elements: 1 2 3 4 5 6
1 2 3 4 5 6
1 2 5 4 3 6
```

2. For each of the program inputs below write the output of the program.

```
import java.util.Scanner;

public class Exam3Trace2 {

    public static void doOne() {
        char[][] pic = new char[6][6];
        for (int i = 0; i < 6; i++)
           for (int j = 0; j < 6; j++) {
              if (i == j || i == 0 || i == 5)
                 pic[i][j] = '*';
              else
                 pic[i][j] = '.';
           }

        for (int i = 0; i < 6; i++) {
           for (int j = 0; j < 6; j++)
              System.out.print(pic[i][j]);
           System.out.println();
        }
        System.out.println();
    }

    public static void doTwo(int n) {
        int[] A = new int[n];
        for (int i = 0; i < n; i++) {
           A[i] = i + 1;
        }

        for (int j = 0; j < 3; j++) {
           int temp = A[0];
           for (int i = 1; i < n; i++) {
              A[i - 1] = A[i];
           }
           A[A.length - 1] = temp;
        }

        for (int i = 0; i < n; i++)
           System.out.print(A[i] + " ");
        System.out.println();
        System.out.println();
    }

    public static void doThree(int n, int k) {
        char[][] pic = new char[6][6];

        for (int i = 0; i < 6; i++)
           for (int j = 0; j < 6; j++) {
              if (i == n || j == k)
                 pic[i][j] = '*';
              else
                 pic[i][j] = '.';
           }

        for (int i = 0; i < 6; i++) {
           for (int j = 0; j < 6; j++)
              System.out.print(pic[i][j]);
           System.out.println();
        }
        System.out.println();
    }
```

```
    public static void main(String[] args) {
        Scanner keyboard = new Scanner(System.in);
        System.out.print("Input: ");
        for (int i = 0; i < 3; i++) {
            switch (keyboard.nextInt()) {
            case 1:
                doOne();
                break;
            case 2:
                doTwo(keyboard.nextInt());
                break;
            case 3:
                doThree(keyboard.nextInt(), keyboard.nextInt());
                break;
            default:
                doOne();
            }
        }
    }
}
```

(a)

```
Input: 2 10 1 3 2 3

4 5 6 7 8 9 10 1 2 3

******
.*....
..*...
...*..
....*.
******

...*..
...*..
******
...*..
...*..
...*..
```

3. Write the output of the program in the box on the next page.

MyStuff.java:

```
public class MyStuff {
   private int num1;
   private double num2;
   public int num3;

   public MyStuff() {
       num1 = 3;
       num2 = 7.5;
       num3 = 5;
   }

   public MyStuff(int a, int b, double c) {
       num1 = a;
       num2 = c;
       num3 = b;
   }

   public void setAll(int a, int b, double c) {
       num1 = a;
       num2 = c;
       num3 = b;
   }

   public void copy(MyStuff m) {
       num1 = m.num1;
       num2 = m.num2;
       num3 = m.num3;
   }

   public void printMyStuff() {
       System.out.println(num1 + "  " + num2
          + "  " + num3);
   }

   public int num1num2() {
       return num1 * (int) num2;
   }

   public int num1num3() {
       return num1 * num3;
   }

   public void setNum1(int a) {
       num1 = a;
   }

   public void setNum2(double a) {
       num2 = a;
   }

   public void setNum3(int a) {
       num3 = a;
   }

   public boolean isEqual(MyStuff m) {
       boolean retval = false;
       if (num1 == m.num1 && num2 == m.num2 &&
             num3 == m.num3)
          retval = true;

       return retval;
   }

   public boolean isGreater(MyStuff m) {
       if (num1 > m.num1)
          return true;
       else if (num1 == m.num1) {
          if (num2 > m.num2)
             return true;
          else if (num2 == m.num2) {
             if (num3 > m.num3)
                return true;
          }
       }
       return false;
   }

   public void mixItUp() {
       num2 = (num1 + num2) / 2.0;
       num1 = 2 * num1 + --num3;
       num3 += 3;
   }
}
```

Main Program:

```
public class Exam3Trace3 {

   public static void printBar(){
       System.out.println("-----");
   }

   public static void main(String[] args) {
       MyStuff m = new MyStuff();
       MyStuff n = new MyStuff(-2, 4, 8.2);
       MyStuff k = new MyStuff(2, 1, 5.3);

       m.printMyStuff();
       n.printMyStuff();
       k.printMyStuff();
       printBar();

       m.num3 = 4;
       n.setNum1(3);
       m.printMyStuff();
       n.printMyStuff();
       System.out.println(m.isEqual(n));
       System.out.println(n.isGreater(m));
       printBar();

       k.mixItUp();
       k.printMyStuff();
       printBar();

       m.setAll(7, 2, 5);
       n.setAll(1, 2, 3.7);
       k.setAll(4, 2, 3.7);

       MyStuff[] A = new MyStuff[3];
       A[0] = k;
       A[1] = m;
       A[2] = n;

       for (int i = 0; i < 2; i++){
          if (A[0].isGreater(A[1])){
             MyStuff temp = A[0];
             A[0] = A[1];
             A[1] = temp;
          }

          if (A[1].isGreater(A[2])){
             MyStuff temp = A[1];
             A[1] = A[2];
             A[2] = temp;
          }
       }

       for (int i = 0; i < 3; i++)
          A[i].printMyStuff();

       printBar();

       A[0] = k;
       A[1] = m;
       A[2] = n;
       for (int i = 0; i < 3; i++)
          A[i].printMyStuff();

       printBar();

       m.setNum3(25);
       for (int i = 0; i < 3; i++)
          A[i].printMyStuff();

       printBar();

       n.copy(k);
       System.out.println(n == k);
    }
 }
```

```
3  7.5  5
-2  8.2  4
2  5.3  1
-----
3  7.5  4
3  8.2  4
false
true
-----
4  3.65  3
-----
1  3.7  2
4  3.7  2
7  5.0  2
-----
4  3.7  2
7  5.0  2
1  3.7  2
-----
4  3.7  2
7  5.0  25
1  3.7  2
-----
false
```

Part 3: Coding (10 Points Each)

1. Write a function that takes a one-dimensional array of doubles as its only parameter and returns the average of the elements in the array. If the array has no elements in it the function should return 0.

```
public static double avg(double[] A) {
   if (A.length == 0)
     return 0;

   double sum = 0;
   for (int i = 0; i < A.length; i++)
     sum += A[i];

   return sum / A.length;
}
```

2. Write a function that takes a one-dimensional array of integers as its only parameter and uses the selection sort to sort the array. Recall that the idea behind the selection sort is to find the largest element in the array and interchange it with the entry in the last position. Then find the largest entry in the array up to the next to last position and interchange it with the entry in the next to last position. and so on until the array is sorted.

```
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. Write a function that will take in a two-dimensional array of integers as its only parameter and return another two-dimensional array of integers that is the transpose of the array. The transpose of an array is where the rows are columns and the columns are the rows. That is, row 1 of the array is column 1 of the transpose, row 2 of the array is column 2 of the transpose, and so on.

```
public static int[][] transpose(int[][] A) {
   int[][] T = new int[A[0].length][A.length];

   for (int i = 0; i < A.length; i++)
      for (int j = 0; j < A[0].length; j++)
         T[j][i] = A[i][j];

   return T;
}
```

4. Write a function that will take in a two-dimensional array of integers as its only parameter and return a one-dimensional array of integers that holds the column sums of the input array.

```
public static int[] columnSum(int[][] A) {
   int[] T = new int[A[0].length];

   for (int j = 0; j < A[0].length; j++) {
      int sum = 0;
      for (int i = 0; i < A.length; i++)
         sum += A[i][j];

      T[j] = sum;
   }
   return T;
}
```

5. Write a function that takes a one-dimensional array of integers and an integer target value and does a binary search for the target value in the array. The return value is to be the position of the target value in the array and -1 if the target value is not in the array.

```
static int binarySearch(int[] A, int N) {
   int lowestPossibleLoc = 0;
   int highestPossibleLoc = A.length - 1;
   while (highestPossibleLoc >= lowestPossibleLoc) {
      int middle = (lowestPossibleLoc + highestPossibleLoc) / 2;
      if (A[middle] == N) {
         return middle;
      } else if (A[middle] > N) {
         highestPossibleLoc = middle - 1;
      } else {
         lowestPossibleLoc = middle + 1;
      }
   }
   return -1;
}
```