1 Short Answer/Method Creation (10 Points Each)

1. Write a method that takes in a one-dimensional array of integers as its only parameter and sorts it using bubble sort.

   Solution:

```java
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;
            }
        }
    }
}
```

2. Write a method that takes in a one-dimensional array of integers and an integer target value as its only two parameters. The method is to do the linear search for the target value in the array. If the target value is found the method should return the index of the target value and if the element is not found then −1 should be returned by the method.

   Solution:

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

3. Write a method that will take in a single integer parameter `sz`, create an array of integers of size `sz`, populate that array with random integers between −100 and 100, and finally return that array.

   Solution:

```java
public static int[] createArray(int sz) {
    int A[] = new int[sz];
    for (int i = 0; i < A.length; i++)
        A[i] = (int) (Math.random() * 201) - 100;

    return A;
}
```

4. Write a method that takes in as a single parameter an ArrayList of integers and returns the average of the array elements.

   Solution:

```java
public static double avgArrayList(ArrayList<Integer> A) {
    double sum = 0;
    for (int i = 0; i < A.size(); i++)
        sum += A.get(i);

    return sum / A.size();
}
```

5. Write a method that will take in a single two-dimensional array of integers as a parameter and flip it vertically. So an input of the array

6	2	3	7	2	6	3
8	8	9	2	6	6	4
3	6	4	3	9	7	8
3	6	1	9	10	4	4
9	8	1	4	3	1	3

will produce an output of the array

```
9    8    1    4    3    1    3
3    6    1    9   10    4    4
3    6    4    3    9    7    8
8    8    9    2    6    6    4
6    2    3    7    2    6    3
```

This method should manipulate the array that is input as a parameter and not output anything, that is, the return type should be void.

Solution:

```
public static void flip(int A[][]) {
    for (int i = 0; i < A.length / 2; i++) {
        for (int j = 0; j < A[0].length; j++) {
            int temp = A[i][j];
            A[i][j] = A[A.length - i - 1][j];
            A[A.length - i - 1][j] = temp;
        }
    }
}
```

6. Write a method that takes in a single parameter of a string, converts the string to all uppercase, removes all spaces, removes all duplicate letters of the string, and finally returns the new string.

Solution:

```
public static String removeDuplicates(String s) {
    s = s.toUpperCase();
    s = s.replaceAll(" ", "");
    String ret = "";

    for (int i = 0; i < s.length(); i++)
        if (ret.indexOf(s.charAt(i)) == -1)
            ret += s.charAt(i);

    return ret;
}
```

7. Write a method that takes in a two-dimensional array of integers and an integer target value as its only two parameters. The method is to do a linear search for the target value in the array. If the target value is found the method should return a one-dimensional array containing the row index in the first cell and the column index in the second cell. If the element is not found then the returned one-dimensional array should have −1 in both cells. The method only needs to find the position of one occurrence of the target, not all of them.

Solution:

```
public static int[] search(int A[][], int t) {
    int pos[] = new int[2];
    for (int i = 0; i < A.length; i++)
        for (int j = 0; j < A[0].length; j++)
            if (A[i][j] == t) {
                pos[0] = i;
                pos[1] = j;
                return pos;
            }

    pos[0] = -1;
    pos[1] = -1;
    return pos;
}
```

2 Program Trace (15 Points)

For each input show the program output.

```
import java.util.Scanner;

public class Exam3TraceS19 {

    public static void printArray(int A[], int s) {
        for (int i = 0; i < s; i++)
            System.out.print(A[i] + " ");
        System.out.println();
    }

    public static void printArray(int A[][], int r,
         int c) {
        for (int i = 0; i < r; i++) {
            for (int j = 0; j < c; j++)
                System.out.print(A[i][j] + " ");
            System.out.println();
        }
    }

    public static void justDoIt(int T[],int a,int q) {
        for (int i = 0; i < a; i++)
            T[i] = a * i + q;
    }

    public static void justDoIt(int T[]) {
        int temp = T[0];
        for (int i = 1; i < 3; i++)
            T[i - 1] = T[i];
        T[2] = temp;
    }

    public static void main(String[] args) {
        Scanner kb = new Scanner(System.in);
        System.out.print("Size: ");
        int s = kb.nextInt();
        System.out.print("Offset: ");
        int u = kb.nextInt();

        int A[] = new int[s];
        int B[] = new int[s];

        justDoIt(A, s, u);
        printArray(A, s);
        printArray(B, s);
        System.out.println("-----");
        B = A;
        A[s / 2] = 10 * s;
        B[0] = s / 2;
        printArray(A, s);
        printArray(B, s);
        System.out.println("-----");
        justDoIt(A, s, u);

        int d = s / 3;
        if (d < s / 3.0)
            d++;

        int C[][] = new int[d][3];
        for (int i = 0; i < s; i++)
            C[i % d][i / d] = A[i];

        printArray(C, d, 3);
        System.out.println("-----");
        if (C[1][0] > 0) {
            printArray(C[1], 3);
            System.out.println("-----");
        }
        int D[] = C[0];
        printArray(D, 3);
        justDoIt(D);
        printArray(D, 3);
        System.out.println("-----");
        printArray(C, d, 3);
    }
}
```

1.
   ```
   Size: 5
   Offset: -10
   ```

   Solution:

   ```
    -10 -5 0 5 10
    0 0 0 0 0
   -----
    2 -5 50 5 10
    2 -5 50 5 10
   -----
   -10 0 10
   -5 5 0
   -----
   -10 0 10
    0 10 -10
   -----
    0 10 -10
   -5 5 0
   ```

2.
   ```
   Size: 7
   Offset: -4
   ```

   Solution:

   ```
    -4 3 10 17 24 31 38
    0 0 0 0 0 0 0
   -----
    3 3 10 70 24 31 38
    3 3 10 70 24 31 38
   -----
   -4 17 38
    3 24 0
    10 31 0
   -----
    3 24 0
   -----
   -4 17 38
    17 38 -4
   -----
    17 38 -4
    3 24 0
    10 31 0
   ```

3 Coding (25 Points)

In this exercise you will create a class structure named `Student` with a constructor and other 6 methods. The data to be stored in the student are their first ans last names as two strings, their student ID as an integer, a two-dimensional array of integers that will hold their scores, and an integer that tracks the number of assignment scores that they have. The two dimensional array is to have 2 rows and 100 columns. Each column will hold the score and worth of one assignment for the student. So if the student has three assignment scores, say a 17 out of 20, a 42 out of 50 and a 78 out of 100 then the array would look like the following and the integer holding the number of scores would have value 3.

17	42	78	0	0	···
20	50	100	0	0	···

Constructor Takes in two strings (first and last name) and an integer ID. Sets the parameter values to the data values in the class.

addScore Takes two integer values, the first is the score the student earned and the second is the value of the assignment. The method is to put these into the correct positions in the array of scores.

printScores Takes no parameters and prints out in a nice aligned two-column format the scores and values in the array.

printNameID Prints out the name of the student in formal form and the ID of the student.

calculatePercentAvergae Calculates the average in percentage form of the student and returns that value as a double.

calculateLetterGrade Calculates the letter grade of the student and returns a string of that letter grade. The letter grade is determined on the 90-80-70-60 scale. That is, 90 and above is an A, 80 to 90 is a B, 70 to 80 is a C, 60 to 70 is a D, and below 60 is an F.

printStudentReport Prints out a student report containing the student's formal name, ID, list of scores, percentage average to 2 decimal places, and their letter grade.

The following is a main program which uses the Student class along with the output of the program.

```java
public class Exam3Code {

    public static void main(String[] args) {
        Student stu = new Student("John", "Smith", 123456789);

        stu.addScore(89, 100);
        stu.addScore(7, 10);
        stu.addScore(16, 20);
        stu.addScore(72, 100);
        stu.addScore(9, 10);
        stu.addScore(5, 10);
        stu.addScore(19, 20);
        stu.addScore(171, 200);

        stu.printNameID();
        stu.printScores();
        System.out.println(stu.calculatePercentAvergae());
        System.out.println(stu.calculateLetterGrade());

        System.out.println();
        System.out.println("----------");
        System.out.println();

        stu.printStudentReport();
    }
}
```

Program Run:

```
Name: Smith, John
Student ID: 123456789
  Score   Value
   89      100
    7       10
   16       20
   72      100
    9       10
    5       10
   19       20
  171      200
82.5531914893617
B

----------

Student Progress Report
Name: Smith, John
Student ID: 123456789

Scores
  Score   Value
   89      100
    7       10
   16       20
   72      100
    9       10
    5       10
   19       20
  171      200

Current Average:  82.55%
Current Letter Garde: B
```

Solution:

```java
public class Student {

    private String firstName;
    private String lastName;
    private int ID;
    private int scores[][] = new int[2][100];
    private int numScores = 0;

    public Student(String fn, String ln, int id) {
        firstName = fn;
        lastName = ln;
        ID = id;
    }

    public void addScore(int s, int v) {
        scores[0][numScores] = s;
        scores[1][numScores] = v;
        numScores++;
    }

    public void printScores() {
        System.out.println("  Score   Value");
        for (int i = 0; i < numScores; i++) {
            System.out.printf("%5d    %5d \n", scores[0][i], scores[1][i]);
        }
    }

    public void printNameID() {
        System.out.println("Name: " + lastName + ", " + firstName);
        System.out.println("Student ID: " + ID);
    }

    public double calculatePercentAvergae() {
        double scoresum = 0;
        double valuesum = 0;

        for (int i = 0; i < numScores; i++) {
            scoresum += scores[0][i];
            valuesum += scores[1][i];
        }

        return scoresum / valuesum * 100;
    }

    public String calculateLetterGrade() {
        double avg = calculatePercentAvergae();

        if (avg >= 90)
            return "A";
        else if (avg >= 80)
            return "B";
        else if (avg >= 70)
            return "C";
        else if (avg >= 60)
            return "D";
        else
            return "F";
    }

    public void printStudentReport() {
        System.out.println("Student Progress Report");
        printNameID();
        System.out.println();
        System.out.println("Scores");
        printScores();
        System.out.println();
        System.out.printf("Current Average: %6.2f%s \n", calculatePercentAvergae(), "%");
        System.out.println("Current Letter Garde: " + calculateLetterGrade());
    }
}
```