Duration of the examination: 1h 15min.
Documentation: Textbook and notes allowed.
Problems to solve: You should solve Problems 1 and 5, and any two problems out of Problems 2, 3, and 4 (I will scale your grade so that it corresponds to a maximum of 100 pts)
Comments: Use your time wisely (no problem deserves that you spend 1/2h on it).
Problem 5 is rather stressful. Avoid keeping it for the last 5 minutes of the exam.

Problem 1 15 Pts
The following blocks of code contain many bugs. Identify as many as you can and indicate which are syntax errors (reported by the compiler), which will produce errors during execution, and which are logical (algorithmic) bugs that will not result in a compilation or execution error, but in an unexpected behavior (for example, an infinite loop).

When deciding if a statement is correct, you should assume that the corrections you made on the previous lines have been applied (i.e. that the preceding lines do not contain errors anymore).

Block 1

```java
int [][]myArray,
[]someArray;
float []A = new float[4][5]
B[3][2] = {{1, 4}, {6, 7}, {4, 4}},
D[][] = new int[5][],
[]E[] = new float[][8];
int [][]F[][];
float [][]G[];

G[0] = B[1];
G = D;
G[1] = H[4];
```
Block 2

```java
int []A[] = new int[4][];
float B[]
   []C = {4.f, 3f, 6.5f, 7.f};
int E[] = {1, 10, 20, 50, 100},
   F[];
int j;

for (int i=0; i<4; i++)
   A[i] = E;

B[0] = C;
C = E;
F = E;

do
   A[j++] = E;
while (j<4);

B = new float[10][];
for (int k=0; k<10; ++k)
{
   B[k] = new float[k+1];
   for (int m=0; m<++k; m++)
      B[k][m] = m+k;
}
Problem 2

Write the code of a method `thresholdedSum` that receives as parameters an array of integers and an integer value (the threshold) and returns the sum of all elements in the array that are strictly greater than the value passed as parameter (obviously, if all elements in the array are smaller than the threshold the method should return 0).

For example, in the code below, `sum1` should receive the value 68 and `sum2` the value 0:

```java
int myArray[] = {23, 12, -4, 5, 8, 5, 12, 21, 5},
    sum1, sum2;

sum1 = thresholdedSum(myArray, 10); // sum1 receives the value 68
sum2 = thresholdedSum(myArray, 30); // sum2 receives the value 0
```
Problem 3

Write the code of a method named `flattenArray` that receives as parameter a (possibly not allocated, and not necessarily rectangular) 2-dimensional array of `double` numbers and returns a 1-dimensional array of `double` containing the elements of the input array re-arranged in one single line.

Examples

If the following array has been declared and initialized

```java
double A[][] = {{10.,20.},{2.,5.,1.},{4.},
                {6.,8.,11.,14.},{7.}};
```

Then the following invocation

```java
double B[] = flattenArray (A);
```

should initialize B to the following value:

```java
{10.,20.,2.,5.,1.,4.,6.,8.};
```

Similarly,

```java
double C[][] = {{},{6.},{},{5.,6.,8.,1.},
                {12.,6.},
                {3.,7.,9.},{5.}};

double D[] = flattenArray (C);
```

should initialize D to the following value:

```java
{6.,5.,6.,8.,1.,12.,6.,3.,7.,9.,5.};
```
Problem 4  

Explain what each of the following two (correct and working) methods accomplish.

```java
public float[][] myFirstMethod(float[][] theArray)
{
    float myArray[][];
    int m = theArray.length,
        n = theArray[0].length;

    myArray = new float[n][m];

    for (int i=0; i<m; i++)
        for (int j=0; j<n; j++)
            myArray[n-j-1][i] = theArray[i][j];

    return myArray;
}

public int mySecondMethod(int A[], int B[])
{
    int m = A.length,
        n = B.length,
        count = 0;
    boolean notFound;

    for (int i=0; i<m; i++)
    {
        notFound = true;
        for (int k=0; (k<i) && notFound; k++)
            if (A[i] == A[k])
                notFound = false;

        if (notFound)
        {
            for (int j=0; (j<n) && notFound; j++)
                if (A[i] == B[j])
                {
                    count++;
                    notFound = false;
                }
        }
    }

    return count;
}
```
Problem 5 25 Pts

Consider the following class implementation (obviously, you wouldn’t want to write anything like it, but it illustrates some mechanisms that you should master).

```java
public class MyClass {
    public int k, m;

    public MyClass() {
        m = 1;
        k = 1;
    }

    public MyClass(int theK, int m) {
        m = m;
        k = ++m + theK;
    }

    public int firstMethod(int m) {
        m += 2;
        k = m + this.k;
        this.m++;
        return m + k;
    }

    public MyClass secondMethod(MyClass theObj) {
        int n;
        n = theObj.firstMethod(this.k) + firstMethod(++m);
        return new MyClass(k+1, n);
    }

    public int thirdMethod(MyClass oneObj, MyClass anotherObj) {
        MyClass myObj, myOtherObj;
        int temp;

        myObj = anotherObj.secondMethod(oneObj);
        myOtherObj = oneObj.secondMethod(this);
        temp = m + oneObj.firstMethod(1) + firstMethod(m);
        return temp;
    }
}
```
What would be the value of instance variables k and m of objects obj1, obj2, and obj3, and of local variables k and n in the main method after the execution of each of the following lines of code.

Note that it is to your advantage in this type of problem to provide information on what you are trying to do, so that I can give you partial credit even if your answer is wrong.

You will find several copies of the source code for MyClass in the appendix.

```java
public static void main(String[] arg)
{
    MyClass obj1, obj2, obj3;
    int k = 0,
        n = 1;
    obj1 = new MyClass();
    obj2 = new MyClass(2, 3);
    k = obj1.firstMethod(n);
    n = obj2.firstMethod(n);

    // Reset 1
    obj1 = new MyClass();
    obj2 = new MyClass(2, 3);
    obj3 = obj2.secondMethod(obj1);
    obj2 = obj2.secondMethod(obj2);

    // Reset 2
    obj1 = new MyClass();
    obj2 = new MyClass(2, 3);
    k = obj2.thirdMethod(obj2, obj1);
}
```
Appendix: Code of class MyClass

```java
public class MyClass {
    public int k, m;

    public MyClass() {
        m = 1;
        k = 1;
    }

    public MyClass(int theK, int m) {
        m = m;
        k = ++m + theK;
    }

    public int firstMethod(int m) {
        m += 2;
        k = m + this.k;
        this.m++;

        return m + k;
    }

    public MyClass secondMethod(MyClass theObj) {
        int n;

        n = theObj.firstMethod(this.k) + firstMethod(++m);

        return new MyClass(k+1, n);
    }

    public int thirdMethod(MyClass oneObj, MyClass anotherObj) {
        MyClass myObj, myOtherObj;
        int temp;

        myObj = anotherObj.secondMethod(oneObj);
        myOtherObj = oneObj.secondMethod(this);

        temp = m + oneObj.firstMethod(1) + firstMethod(m);

        return temp;
    }
}
```
Appendix: Code of class MyClass

public class MyClass {
    public int k, m;

    public MyClass() {
        m = 1;
        k = 1;
    }

    public MyClass(int theK, int m) {
        m = m;
        k = ++m + theK;
    }

    public int firstMethod(int m) {
        m += 2;
        k = m + this.k;
        this.m++;

        return m + k;
    }

    public MyClass secondMethod(MyClass theObj) {
        int n;

        n = theObj.firstMethod(this.k) + firstMethod(++m);

        return new MyClass(k+1, n);
    }

    public int thirdMethod(MyClass oneObj, MyClass anotherObj) {
        MyClass myObj, myOtherObj;
        int temp;

        myObj = anotherObj.secondMethod(oneObj);
        myOtherObj = oneObj.secondMethod(this);

        temp = m + oneObj.firstMethod(1) + firstMethod(m);

        return temp;
    }
}
Appendix: Code of class MyClass

```java
public class MyClass {
    public int k, m;

    public MyClass() {
        m = 1;
        k = 1;
    }

    public MyClass(int theK, int m) {
        m = m;
        k = ++m + theK;
    }

    public int firstMethod(int m) {
        m += 2;
        k = m + this.k;
        this.m++;

        return m + k;
    }

    public MyClass secondMethod(MyClass theObj) {
        int n;

        n = theObj.firstMethod(this.k) + firstMethod(++m);

        return new MyClass(k+1, n);
    }

    public int thirdMethod(MyClass oneObj, MyClass anotherObj) {
        MyClass myObj, myOtherObj;
        int temp;

        myObj = anotherObj.secondMethod(oneObj);
        myOtherObj = oneObj.secondMethod(this);

        temp = m + oneObj.firstMethod(1) + firstMethod(m);

        return temp;
    }
}
```
Appendix: Code of class MyClass

```java
class MyClass {
  public int k, m;

  public MyClass() {
    m = 1;
    k = 1;
  }

  public MyClass(int theK, int m) {
    m = m;
    k = ++m + theK;
  }

  public int firstMethod(int m) {
    m += 2;
    k = m + this.k;
    this.m++;
    return m + k;
  }

  public MyClass secondMethod(MyClass theObj) {
    int n;
    n = theObj.firstMethod(this.k) + firstMethod(++m);
    return new MyClass(k+1, n);
  }

  public int thirdMethod(MyClass oneObj, MyClass anotherObj) {
    MyClass myObj, myOtherObj;
    int temp;
    myObj = anotherObj.secondMethod(oneObj);
    myOtherObj = oneObj.secondMethod(this);
    temp = m + oneObj.firstMethod(1) + firstMethod(m);
    return temp;
  }
}
```
Appendix: Code of class MyClass

```java
public class MyClass {
    public int k, m;

    public MyClass() {
        m = 1;
        k = 1;
    }

    public MyClass(int theK, int m) {
        m = m;
        k = ++m + theK;
    }

    public int firstMethod(int m) {
        m += 2;
        k = m + this.k;
        this.m++;
        return m + k;
    }

    public MyClass secondMethod(MyClass theObj) {
        int n;

        n = theObj.firstMethod(this.k) + firstMethod(++m);
        return new MyClass(k+1, n);
    }

    public int thirdMethod(MyClass oneObj, MyClass anotherObj) {
        MyClass myObj, myOtherObj;
        int temp;

        myObj = anotherObj.secondMethod(oneObj);
        myOtherObj = oneObj.secondMethod(this);
        temp = m + oneObj.firstMethod(1) + firstMethod(m);
        return temp;
    }
}
```