Introduction to Computer Science 1 (2020/2021)
– Resit Exam –

— Do not turn this page before the official start of the exam! —

First Name, Surname: _____

Student ID: _____

Program: Bachelor Data Science and Artificial Intelligence

Course code: KEN1120

Examiner: Dr. Jerry Spanakis and Dr. Enrique Hortal

Date, time: 25th January 2021, 09:30-11:30

Format: Closed book exam

Allowed aids:

- Pens
- Simple (non-programmable) calculator from the DKE-list of allowed calculators

Instructions to students:

- The exam consists of 8 questions on 11 pages (excluding the 1 cover page).
- Fill in your name and student ID number on each page, including the cover page.
- Answer every question at the reserved space below the questions. Use backsides for brainstorming and scratch space. If you run out of space, use the extra blank pages.
- Ensure that you properly motivate your answers.
- Do not use red pens, and write in a readable way. Answers that cannot be read easily cannot be graded and may therefore lower your grade.
- You are not allowed to have a communication device within your reach, nor to wear or use a watch.
- You have to return all pages of the exam. You are not allowed to take any sheets, even blank, home.
- If you think a question is ambiguous, or even erroneous, and you cannot ask during the exam to clarify this, explain this in detail in the space reserved for the answer to the question.
- If you have not registered for the exam, your answers will not be graded, and thus handled as invalid.
- **Success! Break a pencil.**

The following table will be filled by the examiner:

Question:	1	2	3	4	5	6	7	8	Total
Points:	5	8	8	7	7	15	15	15	80
Score:									

Question 1 (5 points)

For the following questions, circle the correct answer. Every correct answer gives +1 point, every wrong answer gives -0.5 points, no answer gives 0 points.

(i) How many asterisks (symbol '*') are printed by the following piece of code?

```
for (int i=3;i<9;i++) {
    System.out.print("*");
}
```

(a) 6
(b) 7
(c) 9
(d) 10
(e) compiler error

(ii) Which of the following is an infinite loop?

(a) `for (;;) //some code here`
(b) `for (int i=10;i>0;i++)  //some code here`
(c) `for (int i=0;;i++)  //some code here`
(d) all of the above
(e) none of the above

(iii) How many basic data types exist in Java?

(a) 2
(b) 4
(c) 8
(d) 16
(e) 32

(iv) What is *byte code* in Java?

(a) block of code written in Java
(b) code generated by a Java Virtual Machine (JVM)
(c) code generated by a Java Compiler
(d) name of a Java source file
(e) another name for the compiler

(v) What will be the result of the method call `square(5)` if the code for the `square` method is the one given below:

```
public static int square(int n) {
    if (n==1)
        return 1;
    else
        return square(n-1) + 2*n - 1;
}
```

(a) 1
(b) 4
(c) 16
(d) 25
(e) infinite run

Question 2 (8 points)
What is type widening and narrowing? How do you do this in Java? Give two examples (one for widening and one for narrowing) of when this is useful/necessary.

Type widening happens when we try putting together (e.g. in an expression or by assignment) two data types that are different but compatible (e.g. float/dobule, int/double). The resulting type is always the most expressive type.

Example: int x = 5; double z = 4; double y = z + x; In this case it's useful, since otherwise we could not proceed with the assignment.

Type narrowing happens when we convert the type of a value of an expression to a new, less expressive, type. That happens through a casting operator.

Example: Math.random returns a long, so when using it, we need to cast the outcome to e.g. an integer int x = (int) (Math.random()*6 +1);

Question 3 (8 points)
Explain briefly what is happening at each step of the "fetch, decode, execute" cycle (or instruction cycle) and which part(s) of the computer is(are) involved in each step.

Fetch instruction: Load to Control Unit (CU) the next instruction from Memory (M). Involving processor and memory Decode instruction: Decode what the instruction is supposed to be doing. Involves processor. Execute instruction: Executes the instruction by sending the relevant signal to the relevant unit. Involves processor and potentially an I/O device or the memory.

Question 4 (7 points)
You are given the following program in Java. What will be printed in the output?

```
import java.util.Arrays;

public class FancyClass {

public static void fancy(int a, int b, int[] list) {
    int temp;
    temp = list[a];
    list[a] = list[b];
    list[b] = temp;
}

public static void main(String[] args) {
    int value1 = 2;
    int value2 = 3;
    int[] list = {1,3,5,7,9};
```

```
    fancy(value1, value2, list);

    System.out.println(value1);
    System.out.println(value2);
    System.out.println(Arrays.toString(list));


     fancy(list[0], list[1], list);

    System.out.println(value1);
    System.out.println(value2);
    System.out.println(Arrays.toString(list));


     fancy(value1, list[value1], list);


    System.out.println(value1);
    System.out.println(value2);
    System.out.println(Arrays.toString(list));

     }
}
```

There are 3 "blocks" of println statements. The first two are executed and printed as follows. The third one is not giving us anything (there is a runtime error in the method call).

2
3

[1, 3, 7, 5, 9]

2
3

[1, 5, 7, 3, 9]

Runtime error: ArrayIndexOutOfBoundsException: 7

Question 5 (7 points)

A student (newbie in programming) is trying to assess whether the following code will work or not. Can you help? If you think the code works as it is, explain shortly why, otherwise point out the syntax error(s) by naming the specific problem(s) and the relevant line(s).

```
1:  public class surprise {
2:        public static void main(String[] args) {
3:          play(1,2,3);
4:          play(1,2);
5:          int x = play(5,6,7);
6:        }
7:
8:        public static void play(int a, int b, int c) {
9:          System.out.println(a+b+c);
10:        }
```

```
11:
12:       public static void play(int a, int b) {
13:        System.out.println(a+b);
14:       }
15:
16:       public static int play(int a, int b, int c) {
17:        return (a+b+c);
18:       }
19:   }
```

This is an example of the use of overloading for methods, i.e. allowing a method to have the same name but different types and number of parameters. While the method declaration in lines 8-10 and lines 12-14 is okay, the method declaration in lines 16-18 changes the type of the method but still keeps 3 parameters, which is not allowed by overloading. So we have an error in these lines. If the method had 4 parameters, it could be allowed to have a different type. Remember that the method call is identified by its name, number of parameters and types of these parameters, so it needs to be unambiguous.

Question 6 (15 points)

Write a method `multiplySquareMatrices` to calculate the dot product of two **square** matrices with the same size that takes as parameters both matrices (integers) and returns the resulting one. The code can assume that this requirement is met (i.e. you do not need to do any check to the dimensions of the matrices). The dot product of two square matrices can be summarized as a summation of (row-column) products as follows.

Considering the matrices A and B as:

$$\mathrm{A} = \begin{bmatrix} a11 & a12 \\ a21 & a22 \end{bmatrix}; \quad \mathrm{B} = \begin{bmatrix} b11 & b12 \\ b21 & b22 \end{bmatrix}$$

The dot product $\mathrm{A}{\cdot}B = \begin{bmatrix} a11*b11 + a12*b21 & a11*b12 + a12*b22 & a11*b13 + a12*b23 \\ a21*b11 + a22*b21 & a21*b12 + a22*b22 & a21*b13 + a22*b23 \end{bmatrix}$

Similarly, for two 3x3 matrices:

$$\mathrm{A} = \begin{bmatrix} a11 & a12 & a13 \\ a21 & a22 & a23 \\ a31 & a32 & a33 \end{bmatrix}; \quad \mathrm{B} = \begin{bmatrix} b11 & b12 & b13 \\ b21 & b22 & b23 \\ b31 & b32 & b33 \end{bmatrix}$$

The dot product $\mathrm{A}{\cdot}B =$

$$\begin{bmatrix} a11*b11 + a12*b21 + a13*b31 & a11*b12 + a12*b22 + a13*b32 & a11*b13 + a12*b23 + a13*b33 \\ a21*b11 + a22*b21 + a23*b31 & a21*b12 + a22*b22 + a23*b32 & a21*b13 + a22*b23 + a23*b33 \\ a31*b11 + a32*b21 + a33*b31 & a31*b12 + a32*b22 + a33*b32 & a31*b13 + a32*b23 + a33*b33 \end{bmatrix}$$

Example 1:

$$\mathrm{A} = \begin{bmatrix} 1 & 2 \\ 2 & 0 \end{bmatrix} \quad \mathrm{B} = \begin{bmatrix} 3 & 1 \\ 1 & 4 \end{bmatrix}$$

$$\mathrm{A}{\cdot}B = \begin{bmatrix} (1*3+2*1) & (1*1+2*4) \\ (2*3+0*1) & (2*1+0*4) \end{bmatrix} = \begin{bmatrix} 5 & 9 \\ 6 & 2 \end{bmatrix}$$

Example 2:

$$\mathrm{A} = \begin{bmatrix} 1 & 0 & 1 \\ 2 & 1 & 2 \\ 3 & 2 & 3 \end{bmatrix}; \quad \mathrm{B} = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 3 \\ 2 & 3 & 4 \end{bmatrix}$$

$$\mathrm{A}{\cdot}B = \begin{bmatrix} 2 & 4 & 6 \\ 5 & 10 & 15 \\ 8 & 16 & 24 \end{bmatrix}$$

Provide your code in this box:

```
import java.util.Arrays;

public class SquareMatrixMultiplication{
    public static void main(String args[]){
        // Input matrices
        int matrixA[][] = {{1,0,1},{2,1,2},{3,2,3}};
        int matrixB[][] = {{0,1,2},{1,2,3},{2,3,4}};
        // dot(A,B) = [[2, 4, 6], [5, 10, 15], [8, 16, 24]]
        //int matrixA[][] = {{1,2},{2,0}};
        //int matrixB[][] = {{3,1},{1,4}};
        // dot(A,B) = [[4,8],[7,4]]

        // Create the output matrix
        int result[][] = multiplySquareMatrices(matrixA, matrixB);
        System.out.println(Arrays.deepToString(result));

    }
    public static int[][] multiplySquareMatrices (int[][] a, int[][] b) {
        int size = a.length;
        int c[][] = new int[size][size];
        // Multiplication
        for(int i=0;i<size;i++)
            for(int j=0;j<size;j++) {
                c[i][j]=0;
                for(int k=0;k<size;k++)
                    c[i][j]+=a[i][k]*b[k][j];
            }
        return c;
    }
}
```

Question 7 (15 points)

Write a method `hasSameElements` that takes as parameters two 1-dimensional arrays of integers and checks if all the elements of the arrays are equal **regardless of their order**. The method returns `true` if all the elements of one matrix appear in the other one and vice-versa or `false` otherwise. Notice that elements value can be repeated more than once.

- *Example 1:*
  arrayA = 0,1,2,3,0;
  arrayB = 1,3,2,0,5;
  The method must return `false` since number 5 is not an element of arrayA.
- *Example 2:*
  arrayA = 0,1,5,2,0;
  arrayB = 1,0,2,0,5;
  The method must return `true` since all the elements in vectorA are also elements of vectorB (notice that there are two elements with value 0 **in both vectors**).

```
class SameElements {
    public static boolean isWithin (int elem, int[] vector) {
        for (int i=0; i<vector.length; i++) {
            if (vector[i] == elem)
                return true;
        }
        return false;
    }

    public static boolean hasSameElements(int[] a, int[] b) {
        // Early stop if it is not feasible
        if (a.length != b.length)
            return false;

        // Check if all the elements in vectorA are in vectorB
        for (int i=0; i<a.length; i++) {
            if (!isWithin(a[i], b))
                return false;
        }
        // Check also if all the elements in vectorB are in vectorA
        for (int i=0; i<b.length; i++) {
            if (!isWithin(b[i], a))
                return false;
        }
        return true;
    }

    public static void main(String[] args) {
        int[] vectorA = {0,1,2,3,0};
        int[] vectorB = {1,3,2,0,5};

        System.out.println(hasSameElements(vectorA, vectorB));

    }
}
```

Question 8 (15 points)

Dutch vaccination plan has been heavily criticised, so the government once again needs your help with a new vaccine. One doctor has isolated the RNA of COVID-19, and as you hopefully know (and remember) from basic biology that is a sequence of nitrogenous bases (adenine (A), guanine (G), thymine (T), and cytosine (C)).

For simplicity reasons we store the sequence of RNA in an integer array where 1 corresponds to A, 2 corresponds to G, 3 corresponds to T and 4 corresponds to C. In order to construct the vaccine, doctors need to re-construct the RNA sequence such that all the same nitrogenous bases appear one after the other in the sequence, i.e. all 1 must be together, all 2 must be together etc. in that particular order, i.e. all 1s come first and all 4s come last.

Write a method called `vaccineRNA` that takes as parameter a 1-dimensional array called RNA and returns a 1-dimensional array that contains the transformed RNA sequence such that all the same nitrogenous bases are together. The original array must remain unchanged.

- *Example 1:*
  RNA = 1,1,2,3,2,4,4,4,1,2,4;
  The method must return the array 1,1,1,2,2,2,3,4,4,4,4.
- *Example 2:*
  RNA = 4,4,4,4,3,3,3,3,2,2,2,2,1,1,1,1;
  The method must return the array 1,1,1,1,2,2,2,2,3,3,3,3,4,4,4,4.

```
import java.util.Arrays;

public class Virus {

    public static int count (int [] b, int j) {
        int count=0;
        for (int i=0;i<b.length;i++) {
            if (b[i]==j) count++;
        }
        return count;
    }

    public static int[] findVaccine(int[] RNA)
    {
        int[] newRNA = new int[RNA.length];

        int pos=0;
        for (int i=1;i<=4;i++) {
            for (int j=0;j<count(RNA,i);j++) {
                newRNA[pos++]=i;
            }

        }
        return newRNA;

    }

    public static void main(String[] args) {

        int[] RNA1 = {1,1,2,3,2,4,4,4,1,2,4};
        int[] RNA2 = {4,4,4,4,3,3,3,3,2,2,2,2,1,1,1,1};

        System.out.println(Arrays.toString(findVaccine(RNA1)));
        System.out.println(Arrays.toString(findVaccine(RNA2)));

    }
}
```