Introduction to Computer Science 1 2020/2021

Exam Questions

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

First name, Surname: **Max Dude**

Student ID: _____

Program: Bachelor Data Science and Knowledge Engineering

Course code: KEN1120

Examiner: dr. Enrique Hortal Quesada and dr. Jerry Spanakis

Date/time: Wednesday, 21st October 2020, 14.00-16.00h

Format: Closed book exam

Allowed aides: Pens, simple (non-programmable) calculator from the DKE-list of allowed calculators.

Instructions to students:

- The exam consists of 8 questions on 15 pages.
- 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
Maximum points:	5	8	8	7	7	15	15	15	80
Achieved points:									

Question 1. (5 Points)

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

I. A variable declared in a Java program, exists in memory for as long as it is determined by its:
a. scope
b. name
c. data type
d. all of the above

II. _____ is a software that converts Java source code to Java bytecode
a. Java API
b. Java compiler
c. Java debugger
d. Java Virtual Machine

III. When you run a Java program, in which part of the computer memory that program is being loaded into?
a. Hard Disk
b. ROM
c. RAM
d. None of the above

IV. For the following piece of code, there is a statement missing in line 3 and is marked as a comment.

```
1  int p=1;
2  for (int i=2; i<=3; i++) {
3  //here we miss a statement
4  }
5  System.out.println(p*(p+1));
```

Which of the following statements needs to be in the position of the comment, so that the last print statement (in line 5) prints 42?

A. `p+=i;` B. `p*=i;`

a. none of the two
b. any of the two
c. B but not A
d. A but not B

V. In a 3-dimensional array (**`myArray)`**, we are storing the amount of precipitation per day where each row represents a week of the year (assuming we have 52 weeks), each column represents a day of the week (from Monday (first column) to Sunday (last column)) and each layer represents a different year (from 1990 (first layer) to 2020 (last layer). If you want to initialize the value of Friday in week 50 from 2000 to 2020 to the value of zero, which is the proper code to be used?

a. for (int i=10; i<=30; i++)
   myArray[4][49][i] = 0;
b. for (int i=10; i<=30; i++)
   myArray[i][4][49] = 0
c. for (int i=11; i<=31; i++)
   myArray[4][49][i] = 0;
d. for (int i=11; i<=31; i++)
   myArray[i][4][49] = 0;
e. None of above

Question 2. (7 points)

Sun Microsystems describes Java as a programming language which is *interpreted*, however in the course we extensively used a *compiler*. Explain shortly why that is.

The Java compilation process takes place in two steps. Firstly, the java source code is compiled into bytecode. Afterwards, the Java Virtual Machine usually interprets this bytecode instead of compile this bytecode into native instructions ready to be run by the CPU.

Question 3. (7 points)

We want to write a method called **`printAndInitialize`** that prints an array which is provided as a parameter and then, initialize all its values to zero and then make a boolean parameter `true` which denotes that all array values are initialized to zero. Check the code below that we prepared for this task.

Is the code syntactically and/or logically correct? Explain in detail your answer by naming the issue that makes the code not correct (syntactically and/or logically).

```
import java.util.Arrays;

public class Question3 {

	public static int[] printAndInitialize (int[] array, boolean init)
{

		System.out.println(Arrays.toString(array));
			if (init) {
				for (int i=0; i<array.length; i++)
					array[i] = 0;
			}

		return array;
	}

	public static void main(String[] args) {
		boolean init = true;
		int[] myArray = {1,2,3,4};

		int[] returnArray = printAndInitialize(myArray, init);

		if ( Arrays.equals(myArray, returnArray) )
			System.out.println("The array was NOT initialized");
		else
			System.out.println("The array was initialized");


}
```

When an array is passed as parameter to a method, the reference is passed as value and therefore, any change performed in the array (input parameter) will affect the original array (the one used in the method's call). Therefore, this code will always return "The array was NOT initialized".

Additionally, there is an ending curly bracket (}) missing for the main method.

Question 4. (8 points)

What will the method below return if called with `n=180`?

```
int fun (int n) {
     if (n<1) return 1;
     else return fun(n/3) + 1;
}
```

Show your work in detail (draw the execution trace/recursive stack).

The result is 6.

fun(180) → return fun(60)+1

→ return fun(20)+1

→ return fun(6)+1

→ return fun(2)+1

→ return fun(0)+1

→ 1 (recursion stops)

Question 5. (8 points)

Suppose we want to write a piece of code that determines the shipping rate for post packages to different parts of the world originating from Maastricht. The rate for Dutch provinces is €5 (unless it's Groningen which is very far so in thtat case it's €10). For the rest of the countries in Europe there is a flat rate of €20. Here is a first attempt:

```
int ShippingRate = 5;
if countrycode.equals("NL")          //countrycode, for Netherlads that is NL
      if statecode.equals("GRO")     //province code, for Groningen it's GRO
      ShippingRate=10;
else
      ShippingRate=20;
```

Is this code syntactically and logically correct? Explain in detail your answer by naming the issue that might potentially cause problems with this code.

The parenthesis in the if statements are required.

Moreover, the "else" statement belong to the second "if" and therefore, the code does not work properly for states in the Netherlands (NL) different than GRO or states in other countries with code GRO. The use of curly brackets is needed here to denote the proper allocation of the "else" statement.

Question 6. (15 points)

The owner of a shop asked you to implement a simpler way of changing the price of her products. She is selling also online so it is very common to need the prices in different currencies (Euro, Dollar and Brazilian real). Moreover, the shop has three different versions of the prices tagged as the regular season, sales and discounted price. Write a piece of code (**`Price`** class) that prints a given array transforming the regular prices in euros (hardcoded, see the **`prices`** array in the initial code provided) following the instructions given by the user using command-line arguments. You must consider the following requirements:

- The user can select the currency using the command -c in the command line followed by one of the available options (namely "euro", "dollar", "real")
- If no currency is specified or the option is incorrect, the values are provided in euros
- The user can select the type of price using the command -p in the command-line followed by one of the available options (namely, "regular", "sales", "discount")
- If no type is specified or the option is incorrect, the values are provided as for the regular season
- The commands can be provided in any order and not all of them are required. Some examples are:
    - `java Price -c dollar -p sales`
    - `java Price -p regular -c euro`
    - `java Price -c real`
    - `java Price -p discount`
- You must start your implementation following the template provided.
- You can assume that at least two command-line arguments will always be provided and that the user will give the arguments in the right order (e.g. -c will always be followed by a currency and -p will always be followed by a price).

```
public class Price {
	static final double CURRENCY_DOLLAR = 1.1708; // 1 EURO = 1.1708 DOLLAR
	static final double CURRENCY_REAL = 6.6275;   // 1 EURO = 6.6275 REAL

	static final double SALES = 0.30;      // Prices during Sales are reduced by 30%
	static final double DISCOUNT = 0.15;   // Discounted prices are reduced by 15%

	public static void main(String[] args) {
		double[] prices = {1, 80.95, 35.95, 6.50, 124.00, 3.25};
```

One possible solution to this question is the use of two nested switch-case statements to firstly identify the command (letter following the symbol "-") and then, check the possible options per commands. The critical part is the use of the array "args" to explore all the commands/options.

Note: if the students use the strings "-c" and "p" instead of checking for the "-" symbol, the code is considered equally valid.

Answer to Question 6 can continue …

```java
import java.util.Arrays;

class Price {
    static final double CURRENCY_DOLLAR = 1.1708;    // 1 EURO = 1.1708 DOLLAR
    static final double CURRENCY_REAL = 6.6275;      // 1 EURO = 6.6275 REAL

    static final double SALES = 0.30;    // Prices during Sales are reduced by 30%
    static final double DISCOUNT = 0.15;     // Discounted prices are reduced by 15%

    public static void main(String[] args) {
        double[] prices = {1, 80.95, 35.95, 6.50, 124.00, 3.25};
        System.out.print(Arrays.toString(prices));


        for (int i=0; i< args.length-1; i++) {
            if (args[i].charAt(0) == '-')
                switch(args[i].charAt(1)) {
                    case 'c':
                        System.out.println("Change of currency");
                        double currency = 1;
                        switch(args[i+1]) {
                            case "dollar":
                                currency = CURRENCY_DOLLAR;
                                break;
                            case "real":
                                currency = CURRENCY_REAL;
                                break;
                            default:
                                break;
                        }
                        for (int j=0; j<prices.length; j++) {
                            prices[j] = prices[j]*currency;
                        }
                        break;
                    case 'p':
                        System.out.println("Change of type");
                        double discount = 0;
                        switch(args[i+1]) {
                            case "sales":
                                discount = SALES;
                                break;
                            case "discount":
                                discount = DISCOUNT;
                                break;
                            default:
                                break;
                        }
                        for (int j=0; j<prices.length; j++) {
                            prices[j] = prices[j]*(1-discount);
                        }
                        break;
                    default: System.out.println("Incorrect command");
                }
        }

        System.out.print(Arrays.toString(prices));
    }

}
```

Question 7. (15 points)

You are asked to write a method **`checkAllSmaller`** that takes as parameter one array of positive integers and returns the number of elements in the array that are smaller or equal than all their subsequent elements.

For example:
With array `A={5,3,7,6,8,6,9}`, your method should return `4`. The bold & underlined elements are the ones that are smaller (or equal) than all their subsequent elements. Note that by default we consider the last element of the array to satisfy the condition (i.e. the last element is always smaller or equal than the element next to it).

With array `B={6,7,4,3,2}`, your method should return `1`. Only the last element satisfies the condition here.

You do not need to do a validity check for the array, i.e. assume that it is always an array of positive integers.

```
public static int checkAllSmaller (int[] arr) {
```

```
    int counter = 0;
    //will be used to count the elements that satisfy the condition

    boolean isGood = true;
    //will be used as a "flag" that is set to false, as soon as we find an
    //element on the right that does not satisfy the property


    //loop for each element of the array
    for (int i=0;i<a.length;i++) {
      isGood=true;  //initialize to true

      //check for all subsequent elements
      for (int j=i;j<a.length && isGood==true;j++) {
          if (a[i]>a[j]) isGood=false;
          //if we find a larger element, then flag becomes false
      }

      if (isGood) counter++;
    }

    return counter;

}
```

Answer to Question 7 can continue …

Question 8. (15 points)

In order to assist the Dutch government with increasing COVID-19 testing, we are helping a medical doctor implement a fast and reliable test. The doctor explained it to us with medical terms but it was too complex, so this is how we understood what the doctor would like to implement: We are given a 2-dimensional rectangular array `MxN` (where `M` is not necessarily equal to `N`) and the elements contain integer values which represent some medical data. In this array, if there is at least one element that has a value equal to the sum of all neighboring element values, then this is the condition that makes the test positive (otherwise it's negative). In this context we consider neighboring elements in all directions: top, bottom, left, right and also diagonally-top-left, diagonally-top-right, diagonally-bottom-left, diagonally-bottom-right, so in total a maximum of 8 elements (can be less than 8 depending on the element in the array). You are asked to write the method **`checkCOVID`** that takes as parameter a 2-dimensional array and returns **`true`** if the test is positive (i.e. the condition above is satisfied) otherwise returns **`false`**.

For example:

with array A= $\begin{bmatrix} 1 & 2 & 3 & 4 \\ 4 & \mathbf{36} & 8 & 5 \\ 5 & 6 & 7 & 7 \\ 0 & \mathbf{19} & 1 & 13 \end{bmatrix}$ your method should return **`true`**, because two elements satisfy the described condition: Element `36` is the sum of all its `8` neighboring elements (`1+2+3+4+8+5+6+7`) and element `19` is the sum of all its `5` neighboring elements (`5+6+7+1+0`).

with array B= $\begin{bmatrix} 1 & 5 \\ 2 & 6 \\ 3 & 7 \\ 4 & 8 \end{bmatrix}$ your method should return **`false`** because no element satisfies the condition.

```
public static boolean checkCOVID (int[][] arr) {
```

The *obvious* solution would to be have an exhaustive check of all boundaries (i.e. take all possible combinations of when (i+1), (j+1), (i-1), (j-1) are within bounds). If you did that (and survived the process, since there are many conditions), then you are getting full points! However, have a look at the next page for a more neat solution.

Answer to Question 8 can continue …

```
    int rows=a.length;
    int cols=a[0].length;

    for (int i=0;i<a.length;i++) {
      for (int j=0;j<a[0].length;j++) {
        int sum=0;
        //check a[i][j] and find which indices exist!
        int startx = Math.max(0,i-1);
        int endx = Math.min(i+1,rows-1);
        int starty = Math.max(0,j-1);
        int endy = Math.min(j+1,cols-1);

        for (int x = startx; x <= endx; x++){
          for(int y = starty; y <= endy; y++){
              if (x != i || y != j){ //make sure that you do not add the current element [i][j]
                sum+=a[x][y];
              } //end if
          }
        }

        if (sum==a[i][j]) return true;
      } //end j-for
    } //end i-for

    return false;
  }
```

Extra answer sheet.

Extra answer sheet.