Review questions - Chapters 1-5

Task 1: Only one answer is correct

* (1P) In Java, variables that cannot change their value are
  1. named as consonants.
  2. actually kept constant by the linker.
  3. identified by the keyword final.
* (1P) Which of the following operators is a binary comparison operator?
  1. "=" Test for equality
  2. ">=" test for "greater than or equal to"
  3. "<<" test on "much bigger"
* (1P) The following applies for switch statements in Java:
  1. The default branch is mandatory.
  2. The case statement should be exited by a break statement, otherwise all of the following cases will be executed.
  3. case-targets must be floating-point literals that can be implicitly cast to double.
* (1P) The following applies for if- statements in Java:
  1. The default branch is mandatory.
  2. In the YES branch, only block statements are allowed, not simple statements.
  3. An existing else- part is always assigned to the last visible if that does not yet have an else.

Task 2: Short question - short answer (1 minute each, 5 points total)

1. Name two basic data types for floating-point numbers in Java.
2. What is the key difference between a *while* and a *do-while* loop?
3. What does the signature of a method consist of?
4. What does the term semantics mean in terms of programming?
5. What is meant by the scope of a local variable?

Task 3: Troubleshooting (15 P)

The following program is intended to scale a double-array f to the value range [0; 1]. To do this, the entire array f is passed into the scaling method. The method creates a new double-array res to store the scaled values in. The res array is returned as a result by the scaling method. The array-variable scaledValues points to the scaled array. This is how the scaling works:

* The smallest value of the input array is a 0 in the res array of the scaling method.
* The largest value of the input array is 1 in the res array of the scaling method.
* The scaling of the other values should be linear.

In the main programme, the newly created field with the scaled values should be output; one value per line.

Find the syntactic errors and improve the semantics of the programme so that it is then an error-free Java programme that fulfills the task set. Try it with pen and paper. Do not rewrite the program!

public Troubleshooting {

public static double scaling(double[] w) {

double[] res = new double[w.length()];  
 double minW = w[0];  
 double maxW = w[0];

for(int i = 1, i < w.length, i++) {  
 if (w[i] > minW) {

minW = w[i];  
 } else {  
 if (w[i] < maxW) {  
 maxW = w[i];  
 }  
 }

}

for(i = 0; i < w.length; i++) {

res[i] = (w[i] - minW) / (minW - maxW);

}

return res;

}  
 public static void main(String args) {

double[] f = { -50.0, 0x23, 25.0, 06, 0.7 };

double[] scaledValues = scaling(f);

System.out.println("\nThe scaled values are:");

for (int i = 0; i < values.length; i++) {

System.out.print(" " + scaledValues[i]);

}

}

}

Task 4: Confusing, isn’t it? (12P)

Specify exactly: What will be output in sequence on the console when the programme below is executed?

public class ConsoleOutput {

public static boolean *isRotated* = false;

public static boolean *isMultiplied* = false;

public static int[] rotate(int[] a, int b) {

int[] t = new int[a.length];

for (int i = a.length-1; i>=0; i--) {

t[i] = *multipliziere*(a[a.length-1-i],b);

}

*isRotated* = true;

return t;

}

public static int multipliziere(int b, int a) {  
 if (a!=1) {  
 *isMultiplied* = true;}

return a \* b;  
 }

public static void main(String[] args) {

int[] b = { 1000, 100, 10, 1 };

b = *rotate*(b,b[b.length-1]);  
 System.*out*.println("Multiply = " + *isMultiplied*);

System.*out*.println("Rotate = " + *isRotated*);

for (int i = 0; i<b.length; i++) {

System.*out*.println("b[" + i + "] = " + b[i]\*4);

}

}

}