Instructions:
This exam is a closed book, no notes, no calculator exam. There are 10 questions total.

For all questions write your answers on the exam and hand in all pages of the exam. There should be 8 pages including this page, please check you have all pages before you begin. If you use the back of the pages for your answers, make sure you indicate that on the front side.

Please write your student number and name at the top of this page, and sign in the space provided.

Note: You may not leave during the last 15 minutes of the exam.

Good luck!

<table>
<thead>
<tr>
<th>Question</th>
<th>Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Question 1</td>
<td>20</td>
</tr>
<tr>
<td>Question 2</td>
<td>10</td>
</tr>
<tr>
<td>Question 3</td>
<td>10</td>
</tr>
<tr>
<td>Question 4</td>
<td>10</td>
</tr>
<tr>
<td>Question 5</td>
<td>10</td>
</tr>
<tr>
<td>Question 6</td>
<td>10</td>
</tr>
<tr>
<td>Question 7</td>
<td>10</td>
</tr>
<tr>
<td>Question 8</td>
<td>10</td>
</tr>
<tr>
<td>Question 9</td>
<td>10</td>
</tr>
<tr>
<td>Total</td>
<td>100</td>
</tr>
</tbody>
</table>

Question 1: (20 marks)
1. The primitive data types DOUBLE and LONG need the same amount of memory (8 Bytes), but LONG covers a smaller range of numbers. Why then sometimes use LONG? (5 marks)
   a. Never use LONG, it's only defined due to compatibility reasons
   b. LONG has a constant integer-precision which is often needed (solution)
   c. It is not possible to store a number without fraction (e.g. 2.0) in a DOUBLE, it is always converted to LONG

2. A method declared as PUBLIC STATIC VOID myMethod() of a class PUBLIC myClass (5 marks)
   a. is visible to all classes (solution)
   b. is a 'class method', i.e. can be used without an instance of myClass by calling myClass.myMethod(); (solution)
   c. is an 'instance method', i.e. an instance of myClass() must be created to access the method.

3. The constructor of a PUBLIC Java-class (5 marks)
   a. is always the only accessible method of a class (solution)
   b. is needed to create instances of the class (solution)
   c. is a good place to initialize data (solution)

4. A Java-class (5 marks)
   a. always contains at least one data element
   b. always contains at least one method
   c. contains either data elements or methods, never both
   d. contains any number of data elements and/or methods (solution)

Question 2: What is the output of the following program (10 marks)

public class TestClass {

    private static int s = 10;
    private int x = 20;
    private int y = 30;

    private void process(int z) {
        x = z + s;
        s = y;
        System.out.println(x);
    }

    public static void main(String args[]) {
        TestClass o1 = new TestClass();
        TestClass o2 = new TestClass();
        o1.process(50);
        o2.process(100);
    }
}
**Question 3**: Write pseudocode that would perform bubble-, selection-sort on an array of integers. (10 marks)

Solution:
```java
void bubbleSort (int[] array) {
    boolean swap;
    do {
        swap = false;
        for (int i = 0; i < array.length - 1; i++) {
            if (array[i] > array[i+1]) {
                int temp = array[i];
                array[i] = array[i+1];
                array[i+1] = temp;
                swap = true;
            }
        }
    } while (swap);
}

void selectionSort (int[] array) {
    for (int i = 0; i < array.length - 1; i++) {
        int minIndex = i;
        for (int j = i + 1; j < array.length; j++) {
            if (array[j] < array[minIndex]) {
                minIndex = j;
            }
        }
        int temp = array[i];
        array[i] = array[minIndex];
        array[minIndex] = temp;
    }
}
```
Question 4: Trace each sorting method. Show the array after each iteration of the outer loop for the first three iterations only. Sort the array into increasing order. Array to start with each time: 80, 70, 50, 60, 40, 30, 10, 20. (10 marks)

Solution:

selection sort

80, 70, 50, 60, 40, 30, 10, 20 start
10, 70, 50, 60, 40, 30, 80, 20 swap 10 and 80
10, 20, 50, 60, 40, 30, 80, 70 swap 20 and 70
10, 20, 30, 60, 40, 50, 80, 70 swap 30 and 50

bubble sort

80, 70, 50, 60, 40, 30, 10, 20 start
70, 50, 60, 40, 30, 10, 20, 80 80 bubbled up
50, 60, 40, 30, 10, 20, 70, 80 70 bubbled up
50, 40, 30, 10, 20, 60, 70, 80 60 bubbled up

Question 5: An array x contains the first 100 non-negative integers in order. So x[i] = i, and i = 0, 1, …, 99. Trace the binary search algorithm using the search value 8. Write the values of the variables low, mid, and high, after each iteration of the algorithm. (10 marks)

Solution:

<table>
<thead>
<tr>
<th>low</th>
<th>mid</th>
<th>high</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>49</td>
<td>99</td>
</tr>
<tr>
<td>0</td>
<td>24</td>
<td>48</td>
</tr>
<tr>
<td>0</td>
<td>11</td>
<td>23</td>
</tr>
<tr>
<td>0</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>10</td>
</tr>
</tbody>
</table>

found at index 8

Question 6: Analyze the running time of the following piece of code (show your work): (10 marks)

```java
int BinarySearch (int[] a, int x)
{
    int left = 0;
    int right = a.length -1;

    while (left<=right)
```
{ int mid = (left+right)/2;
if (a[mid] == x) return mid;
else if (x < a[mid]) right = mid-1;
else left = mid+1;
}
return -1;

Solution: This is a binary search algorithm. Each time through the while loop, the number of elements to be searched is half of what it was in the previous iteration. Therefore the running time is O(log n).

Question 7: Describe in words what the following algorithm does: (10 marks)

```java
int sum(int k)
{
    if (k==1) return 1;
    else return sum(k-1) + k;
}
```

Solution: This recursive algorithm returns the sum from 1 to k. (i.e. \( k + (k-1) + (k-2) + \ldots + 2 + 1 = \frac{k(k+1)}{2} \))
For example, \( \text{sum}(5) = 5 + 4 + 3 + 2 + 1 = 15 \)

Question 8: Write a recursive method called power() that, given an integer n, returns \( 2^n \) (2 to the nth power). Recall that \( 2^0 = 1 \) and \( 2^1 = 2, 2^2 = (2*2) = 4, 2^3 = 2*2*2 = 8 \), etc. power() should not invoke any auxiliary methods. (10 marks)

```java
public int power(int n)
{
    if (n>0)
    return 2 * power(n-1);
else {
    return 1;
}
}
```
Question 9 : Using recursion, complete method `mod(int n,int d)`, which returns the remainder, or modulus, of \( n / d \). Note that we do not imply integer division of \( n \) by \( d \)!
You may not use the `Math` class, division (/), multiplication (*), the modulus operator (%), or any helper methods and classes. Assume that \( n \geq 0 \) and \( d \geq 1 \). (10 marks)

```java
public class Problem3 {
    public static void main(String[] args) {
        System.out.println(mod(11,4)); // output: 3
        System.out.println(mod(10,5)); // output: 0
        System.out.println(mod(1,7)); // output: 1
    }
    // Return remainder of n divided by d. See problem specifications above:
    public static int mod(int n, int d) {
        // Method mod
    }
} // Class

Solution :

{  
    if (n < d)  
        return n;  
    else return  
        mod(n-d,d);  
} // Method mod
```