

SENSORVEILEDNING

Emnekode:	ITF10219
Emnenavn:	Programmering 1
Eksamensform:	Digital eksamen
Dato:	29.11.2019
Faglærer(e):	Georgios Marentakis
Eventuelt:	



Exam consists of 15 subparts that can be grouped in 3 parts.

1st Part (subparts 1-11)

Consists of short programming quizzes with multiple point answers distributed in 11 sections (80 quizzes for a total of 103 points). A correct answer gives one point while a wrong one -0.5 points. Quizzes aim to evaluate the student's understanding of Javascript character set, statements and expressions, types and conversions between them, operators, variables, control statements, arrays, functions, using existing objects, errors handling.

2nd Part (subparts 12-13)

Consists of 2 code reading exercises (23 + 20 = 43 points). These aim to evaluate the ability of the student to read and analyze Javascript code.

3rd Part (subparts 14-15)

Consists of 2 code writing exercises (23 + 31 = 54 points). These are programming exercises which aim to evaluate the ability of the student to write their own code.

Conversion of percentage points to grades

The conversion of points to a specific grade may be done using the following grade limits. However, the examiner may exercise his discretion when assigning a grade based on overall student performance.

A	85
B	75
C	55
D	45
E	35
F	0

Solutions for parts 1-3 follow. For part 1 solutions are final. For parts 2 and 3, it is not necessary that the candidate provides the exact same solution in order to get a full count. Points may be given even if only parts of the logic are correct or there are minor write / code errors.

Description of Grades

A	Fremragende / Outstanding	Outstanding performance that clearly stands out. The candidate shows outstanding command of the subject area and an exceptionally high ability to work independently towards solving problems.
B	Meget god / Very good	Very good performance. The candidate shows very good command of the subject and advanced ability to work independently towards solving problems.
C	God / Good	Overall good performance that is satisfactory in most areas. The candidate's command of the subject area and their ability to work independently towards solving problems is good in the most important areas.
D	Nokså god/ Pretty good	The candidate's command of the subject area and their ability to work independently towards solving problems is acceptable, however, some significant shortcomings are evident.
E	Tilstrækkelig / Sufficient	The candidate's command of the subject area and their ability to work independently towards solving problems meets the bare minimum requirements.
F	Ikke bestått / Fail	Performance that does not meet the minimum academic requirements. The candidate does not demonstrate sufficient command of the subject area nor the ability to work independently towards solving problems.

Appendix: Programming 1 Høst 2019 – Solution Template

Part 1 – Multiple Choice Questions

In the following table the subparts are in columns and the questions in each sub part in columns.

	1	2	3	4	5	6	7	8	9	10	11
1	B	A, B, C, F, G	A, D	C	A	B	D	A	C	C	A
2	A	C	A, B, D	B	D	B	B	B	D	C	A, C, D
3	A, B, E		C, D	B	C	A	B	B	D	D	D
4			A	B	A, B	C	A	A,B,C	B	C	C
5			A, C, D	D	B	B	C	C	A	B	B
6			A, B	A	B	A, C		A	B	A	C
7				C	C	A, B, C, E		A, B	B	A	A
8				B	C			C	C	A	
9				C	A			B	C	D	
10				C				D	A	C	
11								C			

Subpart 12

Subpart 13

The function is an implementation of binary search algorithm. Binary search is used to search a value (x in the function arguments) within a sorted array (y in the function arguments). The function uses a while loop, which will run until the base condition is violated i.e. variable start becomes greater than variable end. Variables start and end are initialized with the index of the beginning and the end of the array. Subsequently the mid index is calculated. If the array length is an odd number, the mid index is $\text{length}(y)/2$; if it is even $\text{length}(y)/2 + 1$ (assuming an integer division). Subsequently, if x equals the array value at the mid index, the function returns true, meaning that the x was found in array y. Otherwise if the array element at the mid index is smaller than x the start index is set to be mid+1, while if the array element at the mid index is greater than x the end is set to be mid-1, equivalent to repeating the search in the right or left half of the array, respectively. The function returns true if x was found in array y and terminates, otherwise it returns false as long as the array is exhausted and start becomes greater than end.

Case 1, the return value is true, Case 2, false, Case 3, false Case 4, true

Subpart 14

```
function fizzBuzz(value) {
  if (value % 15 === 0) { return "FizzBuzz"; }
  if (value % 3 === 0) { return "Fizz"; }
  if (value % 5 === 0) { return "Buzz"; }
  return value;
}
for (var i = 1; i <= 100; i++) {
  console.log(fizzBuzz(i));
}
```

Subpart 15

// intersection(a,b) will return the intersection of a and b arrays

```
function intersection(a,b) {
  let i_a = new Array();
  for (let i=0;i<a.length;i++) {
    for (let j=0;j<b.length;j++) {
      if (b[j]===a[i]) i_a.push(b[j]);
    }
  }
  console.log(i_a);
  return remove_duplicates(i_a);
}
function remove_duplicates(a) {
  let ea = new Array();
  for (let i=0;i<a.length;i++) {
    if (ea.indexOf(a[i])===-1) {
      ea.push(a[i]);
    }
  }
}
```

```
    return ea;  
}
```