

i About the exam



Høgskolen i Østfold

EXAMINATION

Course Code and name:

ITF10219 Programming 1

Date and duration:

29.11.19 , 4 hours

4 A4 pages with notes permitted.

Instructors:

Georgios Marentakis

Please note:

- You are able to view only one question set at the time.
- You may go back to view each question set before you submit your exam.
- You can have the exam question and attachment link (if any) open at the same time.

Part 1-11

Consists of short programming quizzes distributed in 11 sections (80 quizzes for a total of 103 points).

Part 12-13

Consists of 2 code reading exercises (43 points).

Part 14-15

Consists of 2 code writing exercises (54 points).

Grading:

A to F

Results: 20.12.19

Test results will become available in Studentweb.

1 Theory

If a function is defined inside an object as for example m is defined in object o below it is called

```
o = {x:5,y:6, m:function (x) {return x*x}};
```

Select one or more alternatives

- a closure
- a property
- a method
- a literal

If a variable is defined inside an object as x as for example x is defined in object o below it is called

```
o = {x:5,y:6};
```

Select one or more alternatives

- a reference
- a property
- a closure
- a method

Which of the following Javascript types are mutable:

Select one or more alternatives:

- functions
- arrays
- numbers
- strings
- objects

2 DOM

What of the following is correct

Select one or more alternatives

- DOM elements are the target elements for the particular events
- DOM elements can be manipulated by Javascript
- DOM contains the elements that constitute the web page that is currently loaded
- DOM has a tree like structure containing nodes
- Javascript cannot add or remove nodes in the DOM
- DOM is not object-oriented
- The browser can sense when users interact with DOM nodes and fire events

What does the acronym DOM stands for

Select one or more alternatives:

- Document Object Model
- Database Oriented Modelling
- Documentation Objective Mechanism
- Data Object Modelling

3 Expressions/Statements

Which of the following are valid Javascript statements

Select one or more alternatives

- `a=5`
- `-a`
- `a`
- `var sina = Math.sin(a);`

Which of the following is a Javascript expression assuming **a**, **o** have been defined using `var a=5,o={x:6,y:7};`

Select one or more alternatives

- `o.x`
- `[4,5,6]`
- `a=7;`
- `{z:3, y:5}`

Which of the following is a Javascript expression (assuming **a**, **b** has already been defined using `var a,b;`)

Select one or more alternatives

- b-typeof(a)
- a-
- a=3
- a===b

Which of the following is a Javascript expression (assuming a has already been defined using var a;)

Select one or more alternatives

- a^2
- console.log(a)
- a-
- a++

Which of the following is a Javascript expression

Select one or more alternatives

- var a = Math.cos(2*Math.PI);
- 4*Math.cos(Math.PI)
- Math.cos(Math.PI)
- Math.log(4);

Which of the following is a Javascript expression

Select one or more alternatives:

- console.log(1+4);
- 1;
- 1+4;
- var x = 5;

4 Operators

What is the value stored in variable x after the following code has been parsed

```
var x = 0;  
x += 10;
```

Select one or more alternatives

- NaN
- true
- undefined
- 10

What is the value stored in variable c after the following code has been parsed

```
var x = 0, y = 1, c;  
c = Boolean(!x && y);
```

Select one or more alternatives

- false
- undefined
- NaN
- true

What is the value stored in variable x after the following code has been parsed

```
var o = {x:6.2, y:2.1};  
var b = [0, 1, 10];  
x = Boolean(o.x || b[0]);
```

Select one or more alternatives

- false
- NaN
- true
- undefined

What is the value stored in variable x after the following code has been parsed

```
var o = {x:6.2, y:2.1};  
x = Boolean(o);
```

Select one or more alternatives

- true
- Nan
- undefined
- false

What is the value stored in variable x after the following code has been parsed

```
var x = 6.2, y = 2.1, z;  
z = x%y;
```

Select one or more alternatives

- 0
- 2
- 1
- NaN

What is the value stored in variable x after the following code has been parsed

```
var x = 5, y = 10, z = ' dogs';  
Number(x + y + z);
```

Select one or more alternatives

- NaN dogs
- undefined
- 15 dogs
- NaN

What is the value stored in variable z after the following code has been parsed

```
var x = '5', y = '10', z;  
z = Number(x - y);
```

Select one or more alternatives

- 5
- NaN
- 5
- undefined

What is the value stored in variable x after the following code has been parsed

```
var x = 5, y;  
y = Number(x + "1");
```

Select one or more alternatives

- 4
- 51
- 15
- 6

What is the value stored in variable y after the following code has been parsed

```
var x = 5, y;  
y = ++x;
```

Select one or more alternatives

- 7
- 4
- 5
- 6

What is the value stored in variable x after the following code has been parsed

```
var x;  
x = (5+10)*2;
```

Select one or more alternatives:

- 25
- 40
- 10
- 30

5 Variables

What is stored in variable x after the following code is parsed

```
let y = 5, z = 1, x;
```

Select one or more alternatives

- 1
- 5
- undefined
- null

What is stored in variable x after the following code is parsed

```
var x = 5;  
x = 6+14;
```

Select one or more alternatives

- null
- 20
- undefined
- 5

What is stored in variable x after the following code is parsed

```
const x = 5;  
x = 6+13;
```

Select one or more alternatives

- null
- 19
- 5
- undefined

Which of the following is an lvalue

Select one or more alternatives:

- a variable name
- a numeric literal
- an object property
- an object definition literal

What is stored in variable y after the following code is parsed

```
let x=5,y;  
  if (x<3) {  
    let y = 1;  
    y = x*x;  
  }  
}
```

Select one or more alternatives

- undefined
- 25
- 5
- 1

What is stored in variable x after the following code is parsed

```
let x=5;  
function square() {  
  return x*x;  
};  
square();
```

Select one or more alternatives

- 5
- NaN
- 25
- undefined

What is stored in variable y after the following code is parsed

```
let y=5;  
function square_y(x) {  
  y = 4;  
  return x*x*y;  
};  
square(6);
```


Select one or more alternatives

- 16
- 4
- NaN
- 5

What is stored in variable x after the following code is parsed

```
let x;  
function square(x) {  
  x = 4;  
  return x*x;  
};  
x = square(x);
```

Select one or more alternatives

- undefined
- 4
- null
- 16

What is stored in variable x after the following code is parsed

```
let x;  
function square(x) {  
  x = 4;  
  return x*x;  
};
```

Select one or more alternatives

- null
- 16
- 4
- undefined

6 Arrays

What is the value of x after the following code has been parsed

```
var a = [1,2,3,4,5], x;  
x = (a == [1,2,3,4,5]);
```

Select one or more alternatives

- null
- undefined
- true
- false

What is z[3][2] after the following code has been parsed

```
var z=new Array();
for (var i =0; i<5; i++) {
  z[i] = new Array();
  for (var j = 0; j<3; j++){
    z[i][j] = i*j;
  }
}
```

Select one or more alternatives

- 10
- 6
- 8
- 4

What is a[3][1] after the following code has been parsed

```
var a = [[1,2],[3,4],[5,6],[7,8]];
```

Select one or more alternatives

- 8
- 7
- 6
- 1

What is the content of r after the following code has been parsed

```
var a = [4, 2, 4, 5];
var r = (a["1"]==2);
```

Select one or more alternatives

- undefined
- false
- 0
- true

What is the content of a[0] after the following code has been parsed

```
var a = [,2,4,5];
```

Select one or more alternatives

- 5
- undefined
- 0
- 4

Which of the following are valid Javascript array initializations

Select one or more alternatives

- var a = new Array();
- var a = new Array[];
- var a = {5};
- var a = [1,2,3];

Which of the following are valid Javascript array literal

Select one or more alternatives:

- [2 ,, 4]
- [0, 1]
- [{x:2,y:3}, 1, 'a']
- {1,2,3}
- ['a', 1]

7 Errors

What is printed on the console after the following code has been parsed

```
function check(n) {
  if (!(typeof(n) == "number")) {
    throw new TypeError("Not a Number");
  }
}
try {
  check(3);
} catch(error) {
  if (error instanceof TypeError) {
    console.log(error.toString());
  }
} finally {
  console.log("Finally!");
}
```

Select one or more alternatives

- Mistake
- Finally!
- Uncaught Exception
- Not a String

What is printed on the console after the following code has been parsed

```
function check(n) {
  if (!(typeof(x) == "string")) {
    throw new TypeError("Not a String");
  }
}
try {
  check(2000);
```

ITF10219 1 Programming 1

```
    } catch(error) {  
      if (error instanceof TypeError) {  
        console.log(error.toString());  
      }  
    }  
  }  
}
```

Select one or more alternatives

- blah blah
- Type Error: Not a String
- undefined
- Uncaught Exception

What is printed on the console after the following code has been parsed

```
function check(n) {  
  if (!(n >= -500 && n <=500)) {  
    throw new RangeError("blah blah");  
  }  
}  
try {  
  check(2000);  
} catch(error) {  
  if (error instanceof RangeError) {  
    console.log("A Range Error occurred");  
  }  
}
```

Select one or more alternatives

- blah blah
- A Range Error occurred
- undefined
- Uncaught Exception

What is the result after the following code has been parsed

```
throw "Bad String"
```

Select one or more alternatives:

- a Reference Error
- a caught exception
- a Syntax Error
- an uncaught exception

What is the value of variable y after the following code has been parsed

```
function getRectArea(width, height) {  
  if (isNaN(width) || isNaN(height)) {  
    throw "Parameter is not a number!";  
  } else {  
    return width*height;  
  }  
}  
var a,b,y;  
try {  
  getRectArea(a,b);  
} catch (e) {  
  y = e.toString();  
}
```

}

Select one or more alternatives

- NaN
- Parameter is not a number
- Parameter is invalid
- undefined

8 Character Sets

What kind of Javascript literal is 12

Select one or more alternatives

- Object
- Array
- Numeric
- String

What kind of Javascript literal is '12'

Select one or more alternatives

- String
- Object
- Array
- Numeric

Which of the following are valid variable definitions?

Select one or more alternatives

- var this;
- var that;
- var let;
- var switch;

Which of the following are valid variable definitions?

Select one or more alternatives

- var _x;
- var -x;
- var _4x;
- var ø;

How many console lines will be occupied when executing the following

```
console.log("What \n is \n your \t age ");
```

Select one or more alternatives

- 2
- 3
- 1
- 4

Will the following be accepted as a string?

```
"one\  
long\  
line"
```

Select one or more alternatives

- Yes
- No
- It depends

Which of the following is a valid Javascript string literal

Select one or more alternatives

- "abcd
- ""abcd""
- 'abcd'
- ""abcd""

Can you safely omit semicolon if statements are on separate lines when programming Javascript

Select one or more alternatives

- No
- Always
- It depends

Is the following a valid Javascript comment

```
<!-- My Javascript comment -->
```

Select one or more alternatives

- Yes
- No

What kind of Javascript literal is [12]

Select one or more alternatives

- Object
- String
- Numeric
- Array

What kind of Javascript literal is {x:12}

Select one or more alternatives

- Numeric
- Object
- String
- Array

9 Control

What is stored in variable y after the following code is parsed

```
var y;  
for (var i=0,j=10; i < j; i++,j--)  
  y = i+j;
```

Select one or more alternatives

- 5
- 15
- 0
- 10

What is stored in variable y after the following code is parsed

```
var z=-1,y;  
switch (z) {  
  case 1:  
    y = 10;  
  case 2:  
    y = 3;  
  case 4:  
    y = 5;  
  default: 10  
}
```

Select one or more alternatives

- 3
- undefined
- 20
- 5

What is stored in variable y after the following code is parsed

```
var z=2,y;  
switch (z) {  
  case 1:  
    y = 10;  
  case 2:  
    y = 3;  
  case 4:  
    y = 5;
```

```
default: y=10;  
}
```

Select one or more alternatives

- 3
- 5
- 10
- 4

What is stored in variable y after the following code is parsed

```
var z=2,y;  
switch (z) {  
  case 1:  
    y = 10;  
    break;  
  case 2:  
    y = 3;  
    break;  
  case 4:  
    y = 5;  
    break;  
  default: 10  
}
```

Select one or more alternatives

- 3
- undefined
- 4
- 16

What is stored in variable z after the following code is parsed

```
var z=2;  
while (z<10) {  
  z = z*z;  
}
```

Select one or more alternatives

- 16
- undefined
- 8
- 4

What is stored in variable z after the following code is parsed

```
var z;  
for (let i =0; i<20; i+=1) {  
  z = i;  
  if (z == 16) break;  
}
```


Select one or more alternatives

- 5
- 15
- undefined
- 16

What is stored in variable z after the following code is parsed

```
var z;  
for (let i =0; i<3; i+=4) {  
  z = i;  
}
```

Select one or more alternatives

- 8
- undefined
- 0
- 5

What is stored in variable z after the following code is parsed

```
var z;  
for (let i =0; i<10; i+=4) {  
  z = i;  
}
```

Select one or more alternatives

- 5
- 0
- 8
- undefined

What is stored in variable z after the following code is parsed

```
var z;  
for (let i =0; i<19; i++) {  
  z = i;  
}
```

Select one or more alternatives

- 6
- 18
- 5
- undefined

What is stored in variable z after the following code is parsed

```
var x=6, y=12,z;  
if (x<10 && y>10) {  
  z = 5;  
}
```

```
}
```

Select one or more alternatives

- 10
- undefined
- 5
- 6

10 Functions

What will be the value of `y` after the following code is parsed

```
function my_function(a, b) {  
  function square(x) { return x*x; }  
  return square(a) + square(b);  
}  
my_function(4,5);
```

Select one or more alternatives

- 16
- 41
- 25
- 43

What is `y` after the following code is parsed

```
vb = 8;  
function square(x) {  
  x = this.vb;  
  return x*x;  
}  
var y = square(2);
```

Select one or more alternatives

- 128
- 4
- 64
- 16

What will be the value of `y` after the following code is parsed

```
function square(x) {  
  x*x;  
}  
var y = square(2);
```

Select one or more alternatives

- 2
- 4
- 6
- undefined

What will be the value of `y` after the following code is parsed

```
function square(x) {  
  return x*x;  
}  
var y = square(2);
```

Select one or more alternatives

- 6
- 4
- 2
- undefined

What will be the value of `y` after the following code is parsed

```
var y;  
var square1 = function (x) {return x*x;};  
var square2 = square1;  
y = (square1 == square2);
```

Select one or more alternatives

- NaN
- undefined
- true
- false

What will be the value of `y` after the following code is parsed

```
var y;  
var square1 = function (x) {return x*x;};  
var square2 = function (x) {return x*x;};  
y = (square1 == square2);
```

Select one or more alternatives

- NaN
- undefined
- false
- true

What value is stored in `var x` after the following code is parsed

```
function test(x,y) {  
  for (let i=0;i<y;i++) {  
    return i;  
  }  
}
```

```
    }  
  }  
  var x = test(3,4);
```

Select one or more alternatives

- 0
- 1
- 3
- 2

What Javascript type is returned by the following function

```
function test(x,y,z) {  
  return Number(x + y + z);  
}
```

Select one or more alternatives

- an object
- undefined
- a string
- a number

How many parameters does the following function take?

```
function test(x,y,z,b) {  
  return x + y + z + b;  
}
```

Select one or more alternatives:

- 2
- 4
- 3
- 1

What will be the value of y after the following code is parsed

```
function my_function(x) {  
  if (x <= 1) return 1;  
  return x * my_function(x-1);  
}  
let y = my_function(4);
```

Select one or more alternatives

- 24
- 43
- 25
- 16

What is the value of x after the following code has been parsed

```
o = {x:5, y:6};  
let x = o.z;
```

Select one or more alternatives

- NaN
- 11
- Reference Error
- undefined

Which of the following are valid ways to access the property x of an object referenced by variable o

Select one or more alternatives

- o.x
- o\$x
- o["x"]
- o[x']

What is the value of z after the following code has been parsed

```
var z;  
o1 = o{x:5, y:6, m: function (x,y) {return this.x*this.y;}};  
z = o1.m(4,5);
```

Select one or more alternatives

- 30
- 5
- 6
- 20

What is the value of z after the following code has been parsed

```
o = {x:5,y:6, m: function (x,y) {return x*y;}};  
z = o.m(4,5);
```

Select one or more alternatives

- 30
- 6
- 5
- 20

What is the value of z after the following code has been parsed

```
o1 = {x:5, y:6};  
o2 = o1;  
var z = (o1==o2);
```

Select one or more alternatives

- false
- NaN
- true
- undefined

What is the value of z after the following code has been parsed

```
o1 = {x:5,y:6};  
o2 = {x:5,y:6};  
var z = (o1==o2);
```

Select one or more alternatives

- true
- false
- undefined
- NaN

What is the value of z after the following code has been parsed

```
var o = {x:4, y:5};  
let z = o.y;
```

Select one or more alternatives:

- 4
- y
- x
- 5

12 Reading 1

Assuming that variable `ctx` below is a valid 2D Canvas context, use *insert-drawing tool* (pencil tool) to draw what you think will be drawn on the canvas defined below after calling function `draw()`. The picture does not have to be pixel perfect but resemble the outcome as well as possible. Documentation of the methods used in the code is found on the PDF Panel.

TIP: In the RGB color space, if the values of red, green, and blue are the same, the resulting colour will be defined along the grayscale axis and between black (`rgb(0,0,0)`) and white (`rgb(255,255,255)`).

```
<canvas id="canvas" width="300" height="300"> </canvas>
```

```
function draw() {
  var ctx = document.getElementById("canvas").getContext("2d");
  var counter = 0;
  for (var i=0;i<5;i++) {
    for (var j=0;j<5;j++) {
      ctx.fillStyle = "rgb(" + Math.floor(255-42.5*i) + "," + Math.floor(255-42.5*i) + "," + Math.floor(255-42.5*i) +
      ")";
      ctx.strokeRect(25+j*50,25+i*50,50,50);
      ctx.fillRect(25+j*50,25+i*50,50,50);
    }
  }
}
```

Fill in your answer here

13 Reading 2

Describe what is accomplished by the function below. Define what is the function output to the following calls:

1. `f([2,3,4,5],5)`;
2. `f([2,3,4,5],9)`;
3. `f(['abcd'],'a')`
4. `f(['a','b','c','d'],'a')`

```
function f (y, x) {
  let start=0, end=y.length-1;
  while (start<=end) {
    let mid=Math.floor((start + end)/2);
    if (y[mid]==x) {
      return true;
    } else if (y[mid] < x) {
      start = mid + 1;
    } else {
      end = mid - 1;
    }
  }
  return false;
}
```

Fill in your answer here

14 Programming 1

Write a program that uses console.log to print all the numbers from 1 to 100, with three exceptions:

1. for numbers divisible by 3, print "Fizz" instead of the number
2. for numbers divisible by 5 (and not 3), print "Buzz" instead of the number
3. for numbers that are divisible by both 3 and 5 print "FizzBuzz" (and still print "Fizz" or "Buzz" for numbers divisible by only one of those)

Fill in your answer here

15 Programming 2

Write a Javascript function that takes two one-dimensional (numeric or string) arrays as input and calculates their intersection (See definition and examples below).

If arrays have duplicate elements these must appear only once in the array that is returned by the function. You may write a separate function to remove duplicate elements. You may remove duplicates either from the input arrays or from the output arrays. Both options give full points.

The order in which elements appear in the return array does not matter.

Note: You need not check for input type and dimensionality.

Important: No use of ES6 or higher order functions allowed in this task. You should only use control statements and array or string operations, where necessary.

Intersection: The intersection between two arrays is defined as a new array that contains only the elements that are common to both arrays.

Examples:

1. array 1 is [1, 2, 3, 4, 5, 6] and array 2 is [5, 6, 7, 8, 9]. Their intersection is [5, 6].
2. array 1 is ['aa', 'ab', 'ac', 'dd', 'ee'] and array 2 is ['a', 'ee', 'k', 'z', 'ab']. Their intersection is ['ee', 'ab']
3. array 1 is ['aa', 1, 'ab', 2, 'dd', 'ee', 4] and array 2 ['a', 'ee', 'k', 2, 'z', 'ab']. Their intersection is ['ab', 2, 'ee']

Fill in your answer here

Question 12
Attached



The `CanvasRenderingContext2D.fillRect()` method of the Canvas 2D API draws a rectangle that is filled according to the current `fillStyle`.

This method draws directly to the canvas without modifying the current path, so any subsequent `fill()` or `stroke()` calls will have no effect on it.

Syntax

```
void ctx.fillRect(x, y, width, height);
```

The `fillRect()` method draws a filled rectangle whose starting point is at `(x, y)` and whose size is specified by `width` and `height`. The fill style is determined by the current `fillStyle` attribute.

Parameters

x

The x-axis coordinate of the rectangle's starting point.

y

The y-axis coordinate of the rectangle's starting point.

width

The rectangle's width. Positive values are to the right, and negative to the left.

height

The rectangle's height. Positive values are down, and negative are up.

Syntax

```
Math.floor(x)
```

Parameters

x

A number.

Return value

A number representing the largest integer less than or equal to the specified number.

Description

Because `floor()` is a static method of `Math`, you always use it as `Math.floor()`, rather than as a method of a `Math` object you created (`Math` is not a constructor).

 **Note:** `Math.floor(null)` returns `0`, not a `NaN`.

The `CanvasRenderingContext2D.strokeRect()` method of the Canvas 2D API draws a rectangle that is stroked (outlined) according to the current `strokeStyle` and other context settings.

This method draws directly to the canvas without modifying the current path, so any subsequent `fill()` or `stroke()` calls will have no effect on it.

Syntax

```
void ctx.strokeRect(x, y, width, height);
```

The `strokeRect()` method draws a stroked rectangle whose starting point is at `(x, y)` and whose size is specified by `width` and `height`.


Parameters

- x**
The x-axis coordinate of the rectangle's starting point.
- y**
The y-axis coordinate of the rectangle's starting point.

width
The rectangle's width. Positive values are to the right, and negative to the left.

height
The rectangle's height. Positive values are down, and negative are up.

The `CanvasRenderingContext2D.strokeStyle` property of the Canvas 2D API specifies the color, gradient, or pattern to use for the strokes (outlines) around shapes. The default is `#000` (black).

 For more examples of stroke and fill styles, see [Applying styles and color in the Canvas tutorial](#).

Syntax

```
ctx.strokeStyle = color;  
ctx.strokeStyle = gradient;  
ctx.strokeStyle = pattern;
```

Options

color

A `DOMString` parsed as CSS `<color>` value.

gradient

A `CanvasGradient` object (a linear or radial gradient).

pattern

A `CanvasPattern` object (a repeating image).
