“Questions about web publishing material”

* "AJAX" stands for:

- Advanced JavaScript Annotation Standard
- Asynchronous JavaScript Annotation Standard
- Advanced JavaScript and XML
- Asynchronous JavaScript and XML (Correct)

* In order to add jQuery to a Web-page, a Web developer can either download a copy of the jQuery library or include jQuery from a content delivery network (CDN), like Google. Which one of these two options is the better one?

- Include jQuery from a content delivery network (CDN), like Google (Correct)
- Download a copy of jQuery library

* "In order to launch an AJAX request, a web developer has to instantiate (create) an _____ object"

- String
- XMLHttpRequest (Correct)
- XML
- HttpRequest
- Option 5

* "Which of the following statements represents the basic syntax for jQuery"

- $(selector).action() (Correct)
- $(P).HTML()
- $(Ajax).JSON()
- $(XML).DOM()

* "What is the 'readyState' value that indicates an XMLHTTPRequest request is received"

- 1
- 2 (Correct)
- 3
- 4

* "First parameter of XMLHTTPRequest open method has two values, GET and POST. Which one of these values is the proper choice when a cached file is NOT an option."

- GET
- POST (Correct)

* In order to apply a jQuery event to a paragraph element with an "A1" id, the selector part of the jQuery should look like

- $(“#A1”) (Correct)
- $(p.A1)
- $(A1 #p)
- #("A1 .p")

* Usually the Third parameter of XMLHttpRequest is set to "true", which means the type of the request is asynchronous. What does a "false" value of this parameter mean

- Request will be delayed after the page is loaded
- Request will never ever be lunched
- Request has to wait all other XMLHttpRequest in the same page to be completed
- page will not execute any other scripts (i.e., including other XMLHttpRequest request) until this request is processed and the corresponding server (Correct)

* The proper jQuery selector to select all web page elements is:

- `$(“*”)` (Ccorrect)
- `$(this)`
- `$(‘’)`
- `$(‘this’)`

* `$(‘ul li:first’)` jQuery selector selects:

- First “list element” in the first “unordered list” (Correct)
- First “list element” in each “unordered list”

* The proper response type to receive JSON server response is:

- `responseText` (Correct)
- `responseXML`
- `responseJSON`

* Given a table element with six rows, `$(table tr:even)` jQuery selector selects:

- First, third, and fifth table rows
- Second, fourth, and sixth table rows (Correct)
-

* `$('[src]')` jQuery selector selects:

- All HTML elements with "src" attribute (Correct)
- All HTML elements with "href" attribute
- All HTML anchor "a" elements with "href" attribute
- All HTML image "img" elements with "src" attribute

* "What is the proper JavaScript method to enable dealing with JSON format as regular JavaScript objects"

- Object.parse(...)
- JSON.parse(...) (Correct)
- JSON.convertToObject(...)
- JSON.toObject(...)

* "Which of the following is a proper event of select element (drop-down menu)"

- onClick
- onKeyUp
- onChange (Correct)

* "Which of the following is NOT a form event in jQuery"

- load (Correct)
- blur
- change
- submit

* "The proper jQuery event to attach an event handler when the mouse pointer leaves the HTML element is"

- mouseleave (Correct)
- onMouseLeave
- onmouseleave

* "The proper jQuery event to attach an event handler when a form field loses focus is"

- blur (Correct)
- onBlur
- Lose
- onLose

ITeams

Quiz2_AnswerKey

✓ **Correct** 1/1 Points

1 / 1 pt

Auto-graded

3

"AJAX" stands for

- Advanced JavaScript Annotation Standard
- Asynchronous JavaScript Annotation Standard
- Advanced JavaScript and XML
- Asynchronous JavaScript and XML ✓

✓ **Correct** 1/1 Points

1 / 1 pt

Auto-graded

4

In order to add jQuery to a Web-page, a Web developer can either, download a copy of the jQuery library or include jQuery from a content delivery network (CDN), like Google. Which one of these two options is the better one?

- Include jQuery from a content delivery network (CDN), like google ✓
- Download a copy of jQuery library

Auto-graded

5

In order to lunch an AJAX request, a web developer has to instantiate (create) an _____ object

- String
- XMLHttpRequest ✓
- XML
- HttpRequest
- Option 5

✓ **Correct** 1/1 Points

1 / 1 pt

Auto-graded

6

Which of the following statements represents the basic syntax for jQuery

- $(selector).action() ✓
- $(P).HTML()
- $(Ajax).JSON()
- $(XML).DOM()

✓ **Correct** 1/1 Points

1 / 1 pt
Auto-graded

7

What is the "readyState" value that indicates an XMLHttpRequest request is received

- 1
- 2 ✓
- 3
- 4

✓ **Correct** 1/1 Points

1 / 1 pt
Auto-graded

8

First parameter of XMLHttpReguest open method has two values, GET and POST. Which one of these values is the proper choice when a cached file is NOT an option.

- GET
- POST ✓

✓ **Correct** 1/1 Points

1 / 1 pt
Auto-graded

9

In order to apply a jQuery event to a paragraph element with a "A1" id, the selector part of the jQuery should look like

- (●) $("#A1") ✓
- ( ) $(p.A1)
- ( ) $(A1 #p)
- ( ) #("A1 .p")

✓ **Correct** 1/1 Points

1 / 1 pt
Auto-graded

10

Usually the Third parameter of XMLHttpRequest is set to "true", which means the type of the request is asynchronous. What does a "false" value of this parameter mean

- ( ) Request will be delayed after the page is loaded
- ( ) Request will never ever be lunched
- ( ) Request has to wait all other XMLHttpRequest in the same page to be completed
- (●) page will not execute any other scripts (i.e., including other XMLHttpRequest request) until this request is processed and the corresponding server ✓

✓ **Correct** 1/1 Points

1 / 1 pt

Auto-graded

11

The proper jQuery selector to select all web page elements is

- (●) $("*") ✓
- ( ) $(this)
- ( ) $(*)
- ( ) $("this")

✓ **Correct** 1/1 Points

1 / 1 pt

Auto-graded

12

$(ul li:first) jQuery selector selects

- (●) First "list element" in the first "unordered list" ✓
- ( ) First "list element" in each "unordered list"

✕ **Incorrect** 0/1 Points

0 / 1 pt
Auto-graded

13

The proper response type to receive JSON server response is

- ○ responseText ✓
- ○ responseXML
- ◉ responseJSON

✕ **Incorrect** 0/1 Points

0 / 1 pt
Auto-graded

14

Given a table element with a six rows, $(table tr:even) jQuery selector selects

- ○ First, third, and fifth table rows ✓
- ◉ Second, fourth, and sixth table rows

✕ **Incorrect** 0/1 Points

0 / 1 pt
Auto-graded

15

$("[src]") jQuery selector selects

- ○ All HTML elements with "src" attribute ✓
- ○ All HTML elements with "href" attribute
- ○ All HTML anchor "a" elements with "href" attribute
- ◉ All HTML image "img" elements with "src" attribute

✓ **Correct** 1/1 Points

1 / 1 pt

Auto-graded

16

Assume that the server response is returned as XML, i.e., attached snapshot. The proper JavaScript code to retrieve the grade value is
Hint: let the XML response be stored in XMLDoc variable

```
<XML>
<class>Web Publishing</class>
<stdid>3150422</stdid>
<grade>B+</grade>
</XML>
```

- grade= XMLDoc.getElementsByTagName("grade")[0]
- grade= XMLDoc.getElementsByTagName("grade")[0].nodeVale
- grade= XMLDoc.getElementsByTagName("grade")[0].childNodes[0].nodeValue ✓
- grade= XMLDoc.getElementsByTagName("grade")[0].childNodes[0]

✓ **Correct** 1/1 Points

1 / 1 pt
Auto-graded

17

What is the proper JavaScript method to enable dealing with JSON format as regular JavaScript objects

- Object.parse(...)
- JSON.parse(...) ✓
- JSON.convertToObject(...)
- JSON.toObject(...)

✓ **Correct** 1/1 Points

1 / 1 pt
Auto-graded

18

Which of the following is a proper event of select element (drop-down menu)

- onClick
- onKeyUp
- onChange ✓

✓ **Correct** 1/1 Points

1 / 1 pt
Auto-graded

19

Which of the following is NOT a form event in jQuery

- (●) load ✓
- ( ) blur
- ( ) change
- ( ) submit

✓ **Correct** 1/1 Points

1 / 1 pt
Auto-graded

20

Consider the attached piece of HTML code. What does the "this" keyword refers to:

```
<html>
    <body>
    <form>
        First name:
        <input type="text"
        onkeyup="showHint(this.value)">
    </form>
    </body>
</html>
```

- ( ) The HTML page
- ( ) The form element
- (●) The input element ✓
- ( ) The showHint function

✓ **Correct** 1/1 Points

1 / 1 pt
Auto-graded

21

The proper jQuery event to attach an event handler when the mouse pointer leaves the HTML element is

- (●) mouseleave ✓
- ( ) onMouseLeave
- ( ) onmouseleave

✓ **Correct** 1/1 Points

1 / 1 pt
Auto-graded

22

The proper jQuery event to attach an event handler when a form field loses focus is

- (●) blur ✓
- ( ) onBlur
- ( ) Lose
- ( ) onLose

Question 1: Given the following JavaScript code excerpt, Answer the subsequent questions (5 points) :

```
var webPublishing = new Array();
webPublishing[0] = "Web Publishing";
webPublishing[1] = "DOM";
webPublishing[1] = "JSON";
webPublishing[2] = "AJAX";
webPublishing[3] = undefined;
webPublishing[9] = "xmlHTTPRequest";
```

1. Write proper **forEach** iterator to enumerate "*webPublishing*" array. (2.5 points)

```
webPublishing.forEach(function(entry){
console.log(entry);
});
```

2. Write down the expected output of the **forEach** iterator. (2.5 points)

```
Web Publishing
JSON
AJAX
undefined
xmlHTTPRequest
```

Question 2: Given the following array definition in JavaScript. Answer the subsequent questions. (4 points)

```
var charList = new Array();
charList[3] = "F";
charList[0] = "A";
charList[1] = "B";
charList[2.3] = "E";
charList[2] = "D";
charList[1.5] = "C";
```

1. Write conventional **for** iterator to enumerate "*charList*" array. (2 points)

```
for( i = 0; i<charList.length; i++)
console.log(charList[i]);
```

2. Write down the expected output of the **for** iterator. (2 points)

A
B
D
F

Question 3: Object orientation is a powerful mean of programing languages that allow developers to define their own objects, such that objects may have sub-properties and sub-methods. Considering objects in JavaScript, please answer the following questions. (11 points)

1. Define "WebPublishingBook" object in JavaScript with the following properties [use literal notations] (2 points):

- title: Internet and world wide web how to program
- edition: 4
- author: Deitel
- publisher: Pearson

```
> var WebPublishingBook = {title:
  "Internet and WWW", edition: 4,
  author: "Deitel", publisher:
  "Pearson"};
```

2. Define Book constructor, that allows you to instantiate any book of your interest with the same properties listed in part 1 above. (3 Points)

```
function Book(title, edition, author, publisher){
this.title = title;
this.edition = edition;
this.author = author;
this.publisher = publisher;
};
```

3. Use your defined constructor above to instantiate a anew book object labeled "InfoSecBook" with the following properties: (2 Points)

- title: Cryptography and network security
- edition: 8
- author: William Stallings
- publisher: Prentice Hall

```
var infoSecBook = new Book("Cryptography and Network Security", 8,
                           "William Stallings", "Prentice Hall");
```

4. Let the author property of book function constructor stores variable number of last names of book authors (i.e., array structure). Add print function to the book constructor that enables printing all book information including list of authors. (4 points)

```
Book.prototype.print = function(){
var txt = "";
authflag = Array.isArray(this.author);
if(authflag){
var authList= "";
this.author.forEach(function(entry){
authList+=entry+', ';
});
txt = this.title +'\t' + this.edition +
"\twas written by\t"+ authList +
"\tpublished by:\t"
+ this.publisher;
}else
txt = this.title +'\t' + this.edition +
"\twas written by\t"+ this.author +
"\tpublished by:\t"
+ this.publisher;
console.log(txt);
};
```

Question 4: Given the Book object definition as below, answer the following questions (4 Points)

```
<script>
var infoSecBook ={
title:"Cryptography and Information
Security",
author: "William Stallings",
}
}


</script>
```

- **add a *getter* function to get the author attribute (1 point)**

```
get authorName(){
return this.author;},
```

- **Invoke the *getter* function (1 point)**

```
console.log(infoSecBook.authorName);
```

- **define a *setter* method to set the author attribute (1 point)**

```
set c(x){
this.author = x;}
```

- **Invoke the *setter* method to set the author name to "Jason Andress" (1 point)**

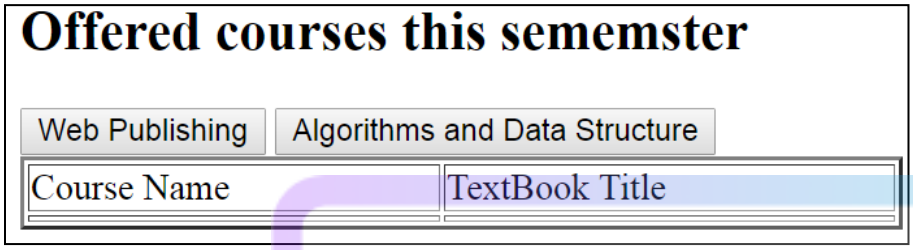
```
infoSecBook.c = "L. S.";

console.log(infoSecBook.authorName);
```

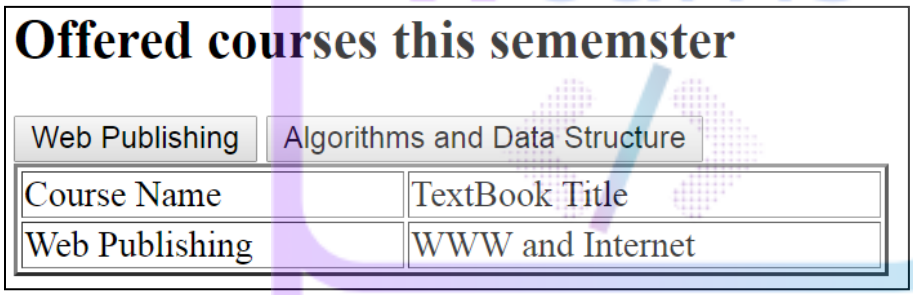
Question 5: Add the necessary JavaScript code to the following Web page, such that when a user clicks on a button, the information of the book of interest will be shown in the table. If the user clicks another button, the new information will overwrite the old one. (6points)

Let book information stored in a JSON object

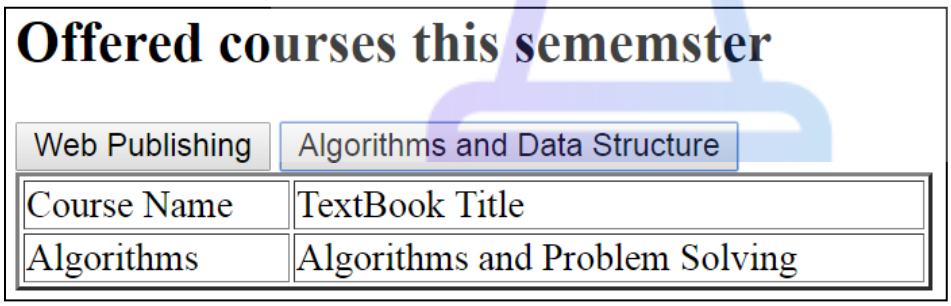
a. Initial view when browser upload the page:



b. View when user clicks "Web Publishing" button



c. View when user clicks "Algorithms and Data Structure" button



```
<!DOCTYPE html>
<html>
<body>
<h2>JSON Object Creation in JavaScript</h2>
<p id="demo"></p>
<button type = "button" onclick = "displyCourse1()">Web Publishing
</button>
<button type = "button" onclick = "displyCourse2()">Algorithms and
Data Structure</button>
<table border= "2" width = "75%">
<tr>
<td>Course Name</td> <td> TextBook Title</td></tr>
<tr>
<td id ="id1"></td><td id = "id2"></td></tr>
</table>
<script>
var textBooks = [{"course":"Web Publishing","textbook":"WWW and
Internet"},{"course":"Algorithms","textbook":"Algorithms and Problem
Solving"}];


function displyCourse1(){
document.getElementById("id1").innerHTML = textBooks[0].course;
document.getElementById("id2").innerHTML = textBooks[0].textbook;
}
function displyCourse2(){
document.getElementById("id1").innerHTML = textBooks[1].course;
document.getElementById("id2").innerHTML = TextBooks[1].textbook;
}
</script>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>
<h2>JSON Object Creation in JavaScript</h2>
<p id="demo"></p>
<button type = "button" onclick =                    >Web Publishing
</button>
<button type = "button" onclick =                    >Algorithms and
Data Structure</button>
<table border= "2" width = "75%">
<tr>
<td>Course Name</td> <td> TextBook Title</td></tr>
<tr>
<td          ></td><td           ></td></tr>
</table>
<script>
var textBooks = [{"course":"Web Publishing","textbook":"WWW and
Internet"},{"course":"Algorithms","textbook":"Algorithms and Problem
Solving"}];


</script>
</body>
</html>
```

Question 1: Answer all of the following questions (10 points):

1. Compare between JSON and XML (4 points)

Criteria	JSON	XML
Tags (need vs. Does NOT need)	Does NOT need	Does need
Arrays (allowed vs NOT allowed)	Does allow	Does NOT allow
Parsing (need vs. Does NOT need	Does NOT need custom parsing	Does need custom parsing
Size (longer vs. shorter)	Is shorter	Is longer

2. What are the types of HTML DOM nodes (3 points)

- Element node
- Attribute node
- Text node

3. Write the required JavaScript to append an image to a webpage(leverage DOM) (3 points)

```
var newImage = document.createElement("IMG");

newImage.setAttribute("src","path-to-the-image-of-interest");

document.body.appendChild(newImage);
```

Question 2: Given the Book object definition as below, answer the following questions (8 Points)

```
<script>
var infoSecBook ={
title:"Cryptography and Information
Security",
author: "William Stallings",
}
}


</script>
```

- **add a *getter* function to get the author attribute (2 points)**

```
get authorName(){
return this.author;},
```

- **Invoke the *getter* function (2 points)**

```
console.log(infoSecBook.authorName);
```

- **define a *setter* method to set the author attribute (2 points)**

```
set c(x){
this.author = x;}
```

- **Invoke the *setter* method to set the author name to "Jason Andress" (2 points)**

```
infoSecBook.c = "L. S.";

console.log(infoSecBook.authorName);
```

Question 6: Given the following simple HTML code. In the script part, write the appropriate code to answer each of the given questions: (5 points)

```
<!DOCTYPE html>
<html>
<body>

<div><h2>Course: Web Publishing</h2></div>
<div><p id="demo">Nothing1</p></div>
<div align = "center"><img src = "Q1_image.jpg"  /></div>
<div><p id="demo">Nothing2</p></div>
<div><p> content </p></div>
<table border = "2" width = "40%">
<tr>
<td>Subject</td>  <td>Hours</td>
</tr>
<tr>
<td>JavaScript Arrays</td>  <td>4.5</td>
</tr>
<tr>
<td>JavaScript Objects</td>  <td>4.5</td>
</tr>
<tr>
<td>DOM</td>  <td>3</td>
</tr>
<tr>
<td>JSON</td>  <td>3</td>
</tr>
<tr>
<td>Ajax</td>  <td>6</td>
</tr>
</table>
```

a. first paragraph given "demo" id

Script	document.getElementById("demo");
Count	1

b. All elements

Script	document.getElemensByTagName("div");
Count	5

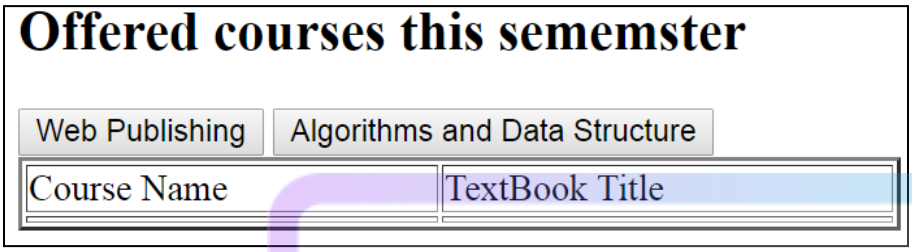
c. All web-page elements

Script	document.getElementsByTagName("*");
Count	---------- -

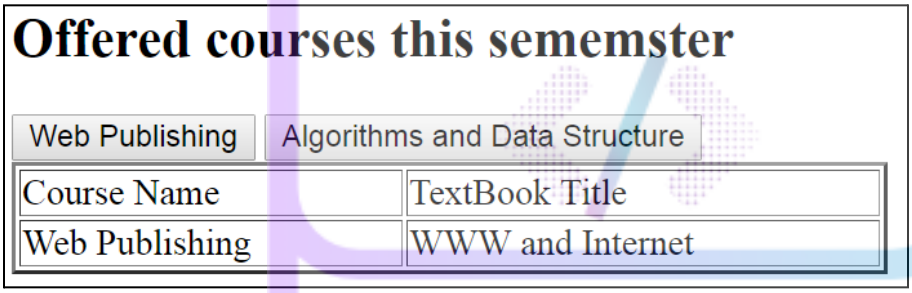
Question 7: Add the necessary JavaScript code to the following Web page, such that when an a user clicks on a button, the information of the book of interest will be shown in the table. If the user clicks another button, the new information will overwrite the old one. (10 points)

Let book information stored in a JSON object

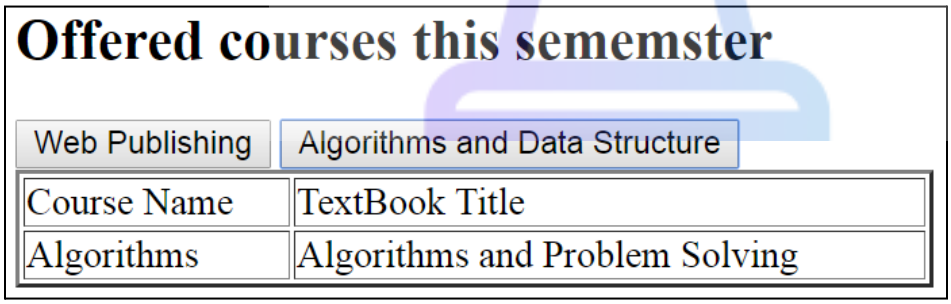
a. Initial view when browser upload the page:



b. View when user clicks "Web Publishing" button



c. View when user clicks "Algorithms and Data Structure" button



```
<!DOCTYPE html>
<html>
<body>
<h2>JSON Object Creation in JavaScript</h2>
<p id="demo"></p>
<button type = "button" onclick = "displyCourse1()">Web Publishing
</button>
<button type = "button" onclick = "displyCourse2()">Algorithms and
Data Structure</button>
<table border= "2" width = "75%">
<tr>
<td>Course Name</td> <td> TextBook Title</td></tr>
<tr>
<td id ="id1"></td><td id = "id2"></td></tr>
</table>
<script>
var textBooks = [{"course":"Web Publishing","textbook":"WWW and
Internet"},{"course":"Algorithms","textbook":"Algorithms and Problem
Solving"}];


function displyCourse1(){
document.getElementById("id1").innerHTML = textBooks[0].course;
document.getElementById("id2").innerHTML = textBooks[0].textbook;
}
function displyCourse2(){
document.getElementById("id1").innerHTML = textBooks[1].course;
document.getElementById("id2").innerHTML = TextBooks[1].textbook;
}
</script>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>
<h2>JSON Object Creation in JavaScript</h2>
<p id="demo"></p>
<button type = "button" onclick =                    >Web Publishing
</button>
<button type = "button" onclick =                    >Algorithms and
Data Structure</button>
<table border= "2" width = "75%">
<tr>
<td>Course Name</td> <td> TextBook Title</td></tr>
<tr>
<td          ></td><td           ></td></tr>
</table>
<script>
var textBooks = [{"course":"Web Publishing","textbook":"WWW and
Internet"},{"course":"Algorithms","textbook":"Algorithms and Problem
Solving"}];
</script>
</body>
</html>
```

Question 1: Given array a declaration as follows, iterate array entries once using for-loop, once again using forEach-function, enumerate the expected output (2 points) :

```
var a= new Array();
a[0] = "A";
a[1] = "B";
a[2]  = "C";
a[2]  = "D"
a[3] = undefined;
a[7]  = "D";
```

For-loop	forEach

//code
For(i = 0; i<a.length; i++)
Console.log(a[i]);
	//code
	a.forEach(function(entry){
	console.log(entry);
	)};

//output A, B, D, Undefined, undefined, undefined, undefined, D	//output A, B, D, undefined, D

Question 2: Assume that you are developing a website for registration, you have been asked to develop an appropriate structure (i.e., object) to instantiate, store and manipulate courses. Answer the following question (12 Points):

Let each course has FOUR attributes, those are:

a) name
b) section
c) classroom
d) instructor

1. Write an appropriate JavaScript code that allow you to instantiate courses of your interest (i.e., define a constructor). (2 points)

```
var course = function(name, section, classroom, instructor){
this.name = name;
this.section = section;
this.classroom = classroom;
this.instructor = instructor;
}
document.write("hello0000000000000000<br />");
var publishing = new course("Web publishing", 3, 208 , "Dr. Sami");
var multimedia = new course("Multimedia", 1, 208, "Dr. Jaber");
```

2. Leverage defined constructor to instantiate any object of your interest. (1 points)

```
12  var publishing = new course("Web Publishing", 1, 208,
    "Ismail Al Taharwa");
13  console.log(publishing);
14  var OS = new course("Operating Systems", 1, 208, "Dr.
    Muaath Abu Faraj");
```

3. Define a function that returns course specifications as a concatenated string of *name* and *section* attributes. such that this function should be invoked as a member method, it still must work for newly instantiated objects. (2 points)

```
course.prototype.displayDetails = function(){
return this.name + '\n' + this.section + '\n' + this.classroom + '\n' + this.instructor;
}
console.log(multimedia.displayDetails());
```

Question 3: Let all courses in Question 2 share the same values for the last three attributes as shown below.

- section : 1
- classroom: 208
- instructor: "Dr. Rami"

You should be able instantiate course objects such that you only need to pass parameters that need to be customized (i.e., *name* attribute).

1. Declare constructor with default values leveraging object literal notation (JSON) (2 points)

The best way is to make a constructor with deafault values:

```
var newCourse = {
name :"Data Structure",
section: 1,
classroom : 208,
instructor: "Dr. Hamzeh",
displyName: function(){
console.log(this.name);
}}

var MLCourse = Object.create(newCourse);
MLCourse.name = "Machine Leraning";
```

2. Instantiate a new course leveraging Object.create() invocation, Let the new course be "Data Structures" set the name property within the Object.create() invocation. (1 points)

```
10  var dataStructures = Object.create(newCourse, {
    name:{value:"DataStructures"}});
```

3. Use both for-in and conventional for iterator to enumerate "Data structure" object (2 points)

For-in	For-loop

//code
`> for(i in DS){ console.log(i + '\t' + DS[i]); };`
`section 1`
`classroom 208`
`instructor dr. Rami`
`display function(){console.log(this.name);}`
	//code
	`props = Object.keys(DS.__proto__);`
	`for(i = 0; i<props.length; i++){ console.log(props[i] + '\t'+DS[props[i]]);}`

//output	//output name Data Structures VM1718:2 section 1 VM1718:2 classroom 208 VM1718:2 instructor dr. Rami VM1718:2 display function(){console.log(this.name);}