Exam 14 May 2018, questions and answers

Fundamentals of Internet Applications Development (City University of Hong Kong)

CITY UNIVERSITY OF HONG KONG

Course code & title	:	CS2204 Fundamentals of Internet Applications Development
Session	:	Semester A 2017/18
Time allowed	:	Two hours

This paper has eight pages (including this cover page).

---

1. There are 4 questions in total.
2. Answer ALL questions.
3. A set of reference pages is provided at the end.

---

*This is a **closed-book** examination.*

No materials or aids are allowed during the whole examination. If any unauthorized materials or aids are found on a candidate during the examination, the candidate will be subject to disciplinary action.

Answer all questions in this paper.

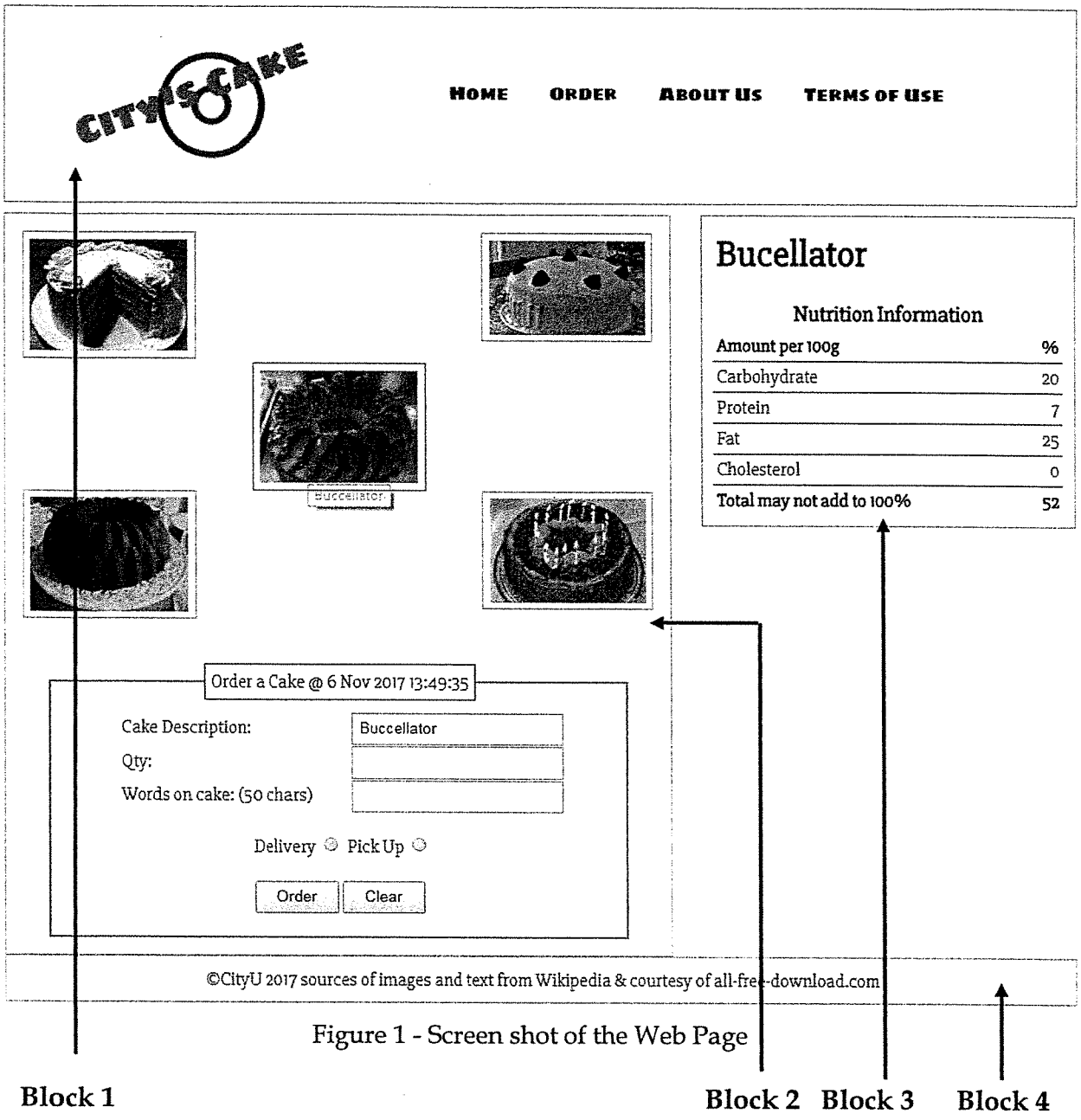

Figure 1 - Screen shot of the Web Page

Question 1 (25%)

Write down only the HTML for the Web page according to the requirements below. You may use empty strings for values that are not given. Include all structural tags (with possible ids) which would enable you to define styles in Question 2.

You need to write HTML elements only for the body part of the Web page. The Web site has 3 folders, *css*, *js* and *image* under the root folder holding style sheets, Javascripts and images respectively. The Web page is in the root folder. Appropriate URLs should be used in your answer.

The Page (Figure 1) consists of 4 main blocks.

Block 1

a) As shown in Figure 1, this is the header block containing the logo and a navigation menu. The logo as a sub-block consists of an image and a 1st level heading ("City's Cake"). The navigation menu is an unordered list with the list items being links. You only need to write HTML for 1 list item and the link destination is not relevant. [4 marks]

Block 2

b) This block contains two sub-blocks. The first one consists of the cake menu and the second one is a form for ordering a cake. [3 marks]

c) The first sub-block contains 5 cake photos, all of them will show the cake description as "bubble text" when pointed by the mouse cursor. You only need to write HTML for 1 photo. [2 marks]

d) The second sub-block is a form as shown in Figure 1. The 1st input (for Cake Description) is read-only, the 2nd input (for Qty) is for number input and the 3rd input can only hold 50 characters maximum. Either "Delivery" or "Pick Up" can be selected, not both. Write HTML for all elements shown. [9 marks]

Block 3

e) The first element is a 1st level heading showing the cake description followed by a table, with "nutrition information" as the title and the first and last rows as header and footer respectively. Write HTML for a full-structured table as shown in Figure 1. [6 marks]

Block 4

f) This block is the footer containing a line of text. [1 mark]

Question 2 (25%)

Add styles to the Web page by writing CSS rules (i.e. no inline styles and no table for layout control) for the HTML in Question 1 according to the requirements below. No need to write styles not asked for, even if they are shown in the screen shot.

Remember to add structural tags, e.g. and identification attributes (e.g. id or class) to your answers in Question 1 or state clearly which and how HTML elements are changed in the answer of this question. You should select elements as specific as possible but at the same time not over using the id attribute.

The Page

a) Use normal flow (static positioning) in the layout of the 4 blocks. Other positioning schemes may be used inside the blocks.

Set margin and padding settings for all elements to 0% and 1% respectively. [1 mark]

Block 1

b) This block is placed at the top of the page. The logo sub-block has a width of 35%, placed at the left. The logo itself is formed by overlapping the heading with the image. The image is 40% in width, centered, within the sub-block. You may use approximate values for the overlapping and angle for placing the heading. [5 marks]

c) The items in the navigation menu are arranged horizontally with no bullet point shown and links are not underlined. [3 marks]

Block 2

d) This block is placed at the left with a width of 60%. [1 mark]

e) Arrange the photos in the 1st sub-block in a "star" form as shown in Figure 1. You can assume, without CSS setting, all photos have the same dimension, 25% in width and 30% in height. You are not allowed to set *id* or *class* attribute for the photos. Approximate values are acceptable. [6 marks]

f) Set the border of the form as shown in Figure 1 to solid line, 1px in width and black in color. The first 3 inputs with their labels as a group, and the delivery/pick up options together with the buttons as another group, are centered within the form. [2 marks]

Block 3

g) Style the table in this block as: the table occupies full width within the block; content in the title, first and last rows are bold; all rows except the last are underlined; content in 1st column is aligned left while 2nd column has its content aligned right. Again, you are not allowed to use *id* or *class* attribute for the table and its descendants. [6 marks]

Block 4

h) The footer must be always on a new line and its content centered. [1 mark]

Question 3 (25%)

Add Javascript to the Web page to achieve the following requirements. You should highlight changes to the HTML codes, if any, in the answer of this question.

a) Write the line of HTML to use a given external Javascript library *lib.js* in the js folder. [1 mark]

Content in this library is as follows:

```
var cakelist=[
{"code":"c01","desc":"Pound Layer","nutrient":[30,20,5,5]},
{"code":"c02","desc":"Strawberry Cream","nutrient":[25,10,51,0]},
{"code":"c03","desc":"Bucellator","nutrient":[20,7,25,0]},
{"code":"c04","desc":"Chocolate","nutrient":[50,30,5,5]},
{"code":"c05","desc":"Birthday","nutrient":[30,40,5,5]}
];

var nutrient=["Carbohydate","Protein","Fat","Cholesterol"];
```

b) Declare and write a function *searchCode(incode)* to search the *cakelist* array containing *cake objects* as defined above. Return the *cake object* when *code* equals to *incode* or *null* when there is no match for *incode*. [4 marks]

c) Declare and write a function *fillInfo(incode)* to fill in content in Block 3 with information from the *cake object* returned from *searchCode(incode)*. The 1st level heading content is from *desc*. The *Code Description* input field in the form in Block 2 is also set with this value.

Content in the 1st column of the table in Block 3 is from the *Array nutrient* in *lib.js* above and the 2nd column information is from *Array nutrient* in the *cake object*. The items in both arrays are in matching order. The total percentage in the last row is calculated by summing up individual percentages. Refer to Figure 1 as example showing the content of Block 2 and 3. [10 marks]

d) Declare and write a function *clickImage()* which will be used as event handler for clicking the photos in Block 2. When a photo is clicked, its border color is changed to red, any previously clicked photo should have its border color set back to silver. Use *fillInfo(incode)* to fill in information in Block 3 and 2. You can also assume *incode* can be obtained from the *id* attribute of the clicked photo, e.g.

```
<img id="c03" src="image/Buccellator.jpg" ...>
```

clickImage() may or may not take parameter, i.e. can be *clickImage(code)* depending on how it is used as event handler. [7 marks]

e) Set up *clickImage()* or *clickImage(code)* as event handler for all the photos in Block 2. You should not assume the number of photos is always 5. [3 marks]

Question 4 (25%)

a) Identify 3 internet applications by naming the application protocol, server and client program. Briefly explain the use of the internet applications. [9 marks]

b) Briefly discuss how the following CSS properties can be used for hiding or showing HTML elements in a Web page. For each of them, explain why page layout and positioning may be affected by their characteristics.

i) display
ii) visibility
iii) z-index [8 marks]

c) Discuss the advantages and limitations of the following ways of setting up event handlers in a Web page.

i) in HTML elements, using event attributes
ii) in Javascript

Give one code example each to illustrate your answers. [8 marks]

~ End ~

CS2204 Fundamentals of Internet Application Development

CSS Reference

Name	Values
'background-color'	<color> | transparent | inherit
'background-image'	<uri> | none | inherit
'background-position'	[[<percentage> | <length> | left | center | right] [<percentage> | <length> | top | center | bottom]?] | [[left | center | right] || [top | center | bottom]] | inherit
'background-repeat'	repeat | repeat-x | repeat-y | no-repeat | inherit
'background'	['background-color' || 'background-image' || 'background-repeat' || 'background-attachment' || 'background-position'] | inherit
'border-color'	[<color> | transparent]{1,4} | inherit
'border-radius'	<length> | <percentage>
'border-spacing'	<length> <length>? | inherit
'border-style'	<border-style>{1,4} | inherit
'border'	[<border-width> || <border-style> || 'border-top-color'] | inherit
'bottom'	<length> | <percentage> | auto | inherit
'box-shadow'	none | <h-shadow> <v-shadow>[blur] [spread] [color] | inherit
'color'	<color> | inherit
'display'	inline | block | list-item | inline-block | table | inline-table | table-row-group | table-header-group | table-footer-group | table-row | table-column-group | table-column | table-cell | table-caption | none | inherit
'float'	left | right | none | inherit
'font-size'	<absolute-size> | <relative-size> | <length> | <percentage> | inherit
'font-style'	normal | italic | oblique | inherit
'left'	<length> | <percentage> | auto | inherit
'height'	<length> | <percentage> | auto | inherit
'list-style-image'	<uri> | none | inherit
'list-style-position'	inside | outside | inherit
'list-style-type'	disc | circle | square | decimal | decimal-leading-zero | lower-roman | upper-roman | lower-greek | lower-latin | upper-latin | armenian | georgian | lower-alpha | upper-alpha | none | inherit
'list-style'	['list-style-type' || 'list-style-position' || 'list-style-image'] | inherit
'margin-right' 'margin-left'	<margin-width> | inherit
'margin-top' 'margin-bottom'	<margin-width> | inherit
'margin'	<margin-width>{1,4} | inherit
'overflow'	visible | hidden | scroll | auto | inherit
'opacity'	number | inherit
'padding-top' 'padding-right' 'padding-bottom' 'padding-left'	<padding-width> | inherit
'padding'	<padding-width>{1,4} | inherit
'position'	static | relative | absolute | fixed | inherit
'text-align'	left | right | center | justify | inherit
'text-decoration'	none | [underline || overline || line-through || blink] | inherit
'text-indent'	<length> | <percentage> | inherit
'text-transform'	capitalize | uppercase | lowercase | none | inherit
'top'	<length> | <percentage> | auto | inherit
'transform'	translate(x,y) | scale(x,y) | rotate(angle)
'vertical-align'	baseline | sub | super | top | text-top | middle | bottom | text-bottom | <percentage> | <length> | inherit
'visibility'	visible | hidden | collapse | inherit
'width'	<length> | <percentage> | auto | inherit

Name	Values
'word-spacing'	normal | <length> | inherit
'z-index'	auto | <integer> | inherit

Javascript Reference

Array Object Properties

Property	Description
constructor	Returns the function that created the Array object's prototype
length	Sets or returns the number of elements in an array
prototype	Allows you to add properties and methods to an Array object

Document / Window / String Object Properties and Methods

Property / Method	Description
document.forms	Returns a collection of all the forms in the document
document.getElementById()	Returns the element that has the ID attribute with the specified value
document.getElementsByName()	Accesses all elements with a specified name
document.getElementsByTagName()	Returns a NodeList containing all elements with the specified tagname
document.URL	Returns the full URL of the document
document.write()	Writes HTML expressions or JavaScript code to a document
document.querySelector()	Returns the first element that match the selector
document.querySelectorAll()	Returns a collection of all elements that meatch the selector
window.localStorage.getItem()	Returns a string/object based on a key/attribute
window.localStorage.setItem()	Sets a string/object based on a key/attribute
window.location.search	Returns the query portion of the URL including the "?" character
substring(start [,end])	Extracts the characters from a string, between two specified indices
split([separator])	Splits a string into an array of substrings