

COMP 322 Internet Systems Fall 2007 Exam 2—Solutions

50 points total

1 (7 pts.). Write a JavaScript statement that assigns to `str1` a string identical to the string in the variable `str` except that every sub-string that consists of one or more digits, a colon (:), then one or more digits has the two strings of digits swapped. For example, if the value of `str` is “12:4 and 6:21”, then “4:12 and 21:6” is assigned to `str1`.

Answer

```
str1 = str.replace( /(\d+):(\d+)/g, "$2:$1" );
```

2 (26 pts.). The following is a listing with gaps of the HTML document whose rendering is shown at right. The entire body of the document is a form. It contains a text box with name `name` (for the user’s name). It also has a field-set with caption “This is your” encompassing three radio buttons, each with name `nametype`. The value for the first is ‘`first`’, the value for the second is ‘`last`’, and the value of the third is ‘`full`’. Another field-set, with caption “How much time do you have now?”, encompasses a drop-down menu named `time` with three alternatives, named ‘`lots`’ (with text `Lots`), ‘`notMuch`’ (with text `Not much`), and ‘`none`’ (with text `None`). Finally, there is a submit button.

Your name:

How much time do you have now?

This is your
 First name
 Last name
 Full name

You must enter your first name and last name, and nothing more.

Done

When this document finishes loading, function `start()` is called. This function assigns a reference to the form to variable `fm`. It then makes the function `verify` the event handler for submit events and makes the function `fastSubmit` the event handler for change events for the drop-down menu. Function `fastSubmit()` checks whether the value for `time` is ‘`none`’; if so, it forces a submission of the form (essentially simulating a click of the submit button).

Function `verify()` copies the value of text box `name` into variable `nm` and searches through the elements of the radio button array `nametype` for the button that is checked; if one is checked, its value is copied to the variable `nmttype`. It then checks whether any of four error conditions hold: (1) no radio button was checked, (2) `nmttype` is ‘`first`’ but the value in the text box does not have the pattern of a single name (an initial uppercase letter followed by one or more lowercase letters), (3) `nmtaype` is ‘`last`’ but the value in the text box does not fit the pattern of a single name, and (4) `nmttype` is ‘`full`’ but the value in the text box does not fit the pattern for a full name (two single names separated by any amount of whitespace). If any of these error conditions holds, `verify()` calls the function `unverified()`, passing it the event that was passed to it and a string containing a message explaining what is wrong.

Function `unverified()` suppresses the submission, makes a new `p` element, assigns the string it was passed as the content of the `p` element, and adds the `p` element to

the end of the body of the document. The screenshot above shows the case where the user indicated a full name but entered only a first name. After an attempt to submit, the text at the bottom appeared.

The gaps in the following listing are labeled with Greek letters. After the listing, these letters are repeated along with a description of the code that should fill the corresponding gaps. You then supply the missing code.

```
<html>
<head>
  <title>Exam 2, Problem 2</title>

  <script type="text/javascript">
    var fm;

    function start()
    {
      fm = α;
      β;
      γ;
    }

    function fastSubmit( event )
    {
      δ
    }

    function verify( event )
    {
      var oneName = ε,
          fullName = ζ,
          nm = η,
          nmtypes = θ,
          ntsize = nmtypes.length,
          nmtype;

      for ( i=0; i<ntsize && ! nmtype; i++ )
        if ( nmtypes[i].checked )
          nmtype = nmtypes[i].value;
    }
  </script>
</head>
<body>
  <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;">
    <input type="text" value="John Doe" />
  </div>
  <input type="submit" value="Submit" />
</body>
</html>
```

Continued

Continued from previous page

```
if ( ! nmtype ) {
    unverified( event, "You must check a radio button." );
}
else if ( nmtype == 'full' && ! nm.match( fullName ) ) {
    unverified( event, "You must enter your first " +
        "name and last name, and nothing more." );
}
else if ( nmtype == 'first' && ! nm.match( oneName ) ) {
    unverified( event, "You must enter your first " +
        "name and nothing more." );
}
else if ( nmtype == 'last' && ! nm.match( oneName ) ) {
    unverified( event, "You must enter your last " +
        "name and nothing more." );
}
}

function unverified( event, msg )
{
    λ
}

</script>
</head>
<body onload="start()">
<form action="prob2.php" method="post">
    <p>Your name: κ_____<p>

    λ

```

Continued

Continued from previous page

u

```
<p style="position: absolute; top: 100; left: 185">
  <input id="sbmt" type="submit" value="Submit">
</p>
</form>
</body>
</html>
```

α (1.5 pts.): A reference to the form element.

Answer: `document.forms[0]`

β (1.5 pts.): Make function `verify` the listener for submit events.

Answer: `fm.addEventListener("submit", verify, false);`

γ (1.5 pts.): Make function `fastSubmit` the listener for change events in the drop-down menu, `time`.

Answer: `fm.time.addEventListener("change", fastSubmit, false);`

δ (4 pts.): Write the body for `fastSubmit()`. This forces a submission (in effect, clicking the submit button) if the value for the drop-down menu, `time`, is 'none'.

Answer:

```
var tm = fm.time.value;
if ( tm == 'none' )
  document.getElementById( "sbmt" ).click();
```

ϵ : A regular expression that matches a name (an uppercase letter followed by one or more lowercase letters) with nothing before or after it.

Answer: This problem was taken off the exam and is not to be graded.

For reference: `/^[A-Z][a-z]+$/`

ζ: A regular expression that matches a full name: a first followed by a last name, that is, an uppercase letter followed by one or more lowercase letters (i.e., the name pattern), then any amount of whitespace, then the name pattern again. There should be no characters before or after this pattern.

Answer: This problem was taken off the exam and is not to be graded.

For reference: `/^[A-Z][a-z]+\s+[A-Z][a-z]+$/`

η (1 pt.): The value of the text box named name

Answer: `fm.name.value`

θ (1 pt.): The array of elements corresponding to the radio buttons named nametype

Answer: `fm.nametype`

ι (6 pts.): Write the body of the function `unverified(event, msg)`, where `event` is an `Event` object and `msg` is a string with an error message. This suppresses the submission, makes a new `p` element, assigns `msg` as the content of the `p` element, and adds the `p` element to the end of the body of the document.

Answer:

```
event.preventDefault();
var newP = document.createElement( "p" );
newP.innerHTML = msg;
document.body.appendChild( newP );
```

κ (1.5 pts): A text box named name whose width is the width of 18 characters

Answer: `<input type="text" name="name" size="18">`

λ (4 pts.): A field-set with caption “This is your”. It is 5 pixels from the top and 220 pixels from the left margin. It contains the three radio buttons shown in the screenshot, all named ‘nametype’. Their values are, from top to bottom, ‘first’, ‘last’, and ‘full’.

Answer:

```
<fieldset style="position: absolute; top: 5; left: 220">
  <legend>This is your</legend>
  <input type="radio" name="nametype" value="first">
    First name<br />
  <input type="radio" name="nametype" value="last">
    Last name<br />
  <input type="radio" name="nametype" value="full">
    Full name
</fieldset>
```

μ (4 pts.): A field-set, with caption “How much time do you have now?”, with a line break between “time” and “do”. It encompasses a drop-down menu named time with three alternatives, named ‘lots’ (with text Lots), ‘notMuch’ (with text Not much), and ‘none’ (with text None). The field-set is 120 pixels wide.

Answer:

```
<fieldset style="width: 120">
  <legend>How much time<br>do you have now?</legend>
  <select name="time" size="1">
    <option value="lots">Lots</option>
    <option value="notMuch">Not much</option>
    <option value="none">None</option>
  </select>
</fieldset>
```

3 (17 pts.). The following is a listing with gaps of a PHP script that first copies the value of the form variables `first` (the user's first name), `last` (his last name), and `age` (his age) into the program variables `$first`, `$last`, and `$age`, respectively. If the user did not fill in the text box for one of these, the corresponding variable will contain the empty string. The script defines constants whose values are defaults for these values. `DEFAULT_NAME` has the value 'Unknown' and `DEFAULT_AGE` has the value 21. The script checks whether `$first` or `$last` is the empty string; if so, sets it to the default name. It also checks whether `$age` is the empty string and, if it is not, whether it cannot be interpreted as a number other than 0 (i.e., whether it coerces to 0); if so, it sets `$age` to the default age.

At right is shown the rendering of the HTML produced by this script when the user supplied the last name 'Smith' and '27' for his age but gave nothing as his first name. The script outputs a paragraph with a greeting that echoes the first and last names. The page is meant to be printed and filled out by the user and then taken to his psychiatrist's office. The left column gives five-year periods, starting at the user's birth, and going up to

Good day, Unknown Smith!

Print the following form, fill in the most significant life experience you had in each of the five-year periods, and bring it to Dr. Schickelgruber's office at your next appointment.

Period	Most significant experience
Years 0-4	
Years 5-9	
Years 10-14	
Years 15-19	
Years 20-24	
Years 25-27	

Done

the present. The last period will be less than five years unless the user's age minus 1 is divisible without remainder by 5. The second column, except for the heading, is blank. The script creates this table with a loop where the loop control variable starts at 0 and is incremented by 5 on each iteration. The tricky part is to get the upper bound on the last period correct. Note that the expression

$$\$i+4 < \$age ? \$i+4 : \$age$$

evaluates to the smaller of $\$i+4$ and $\$age$.

The following is the listing, where gaps are labeled with Greek letters. After the listing, the letters occur along with descriptions of the code that goes in the corresponding gaps. You there supply the missing code.

```
<?php
    α_____i;
    _____i;

    β_____i;
    _____i;
    _____i;

    if ( γ_____ )
        $first = DEFAULT_NAME;

    if ( δ_____ )
        $last = DEFAULT_NAME;

    if ( ε_____ )
        $age = DEFAULT_AGE;
    else
        $age = ζ_____i;
?>
<html>
<head>
    <title>Exam 2, Problem 3</title>
</head>
<body>
    <?php
        η_____i;
    ?>

    <p>
        Print the following form, fill in the most significant
        life experience you had in each of the five-year periods,
        and bring it to Dr. Schickelgruber's office at your next
        appointment.
    </p>

    <table border='1' width='100%'>
        <colgroup span='1' width="20">
        </colgroup>
        <thead>
            <tr><th>Period</th>
                <th>Most significant experience</th></tr>
        </thead>
        <tbody>
```

Continued

Continued from previous page

```
<?php


|          |
|----------|
| <u>θ</u> |
|----------|


?>
</tbody>
</table>
</body>
</html>
```

α (2 pts., 2 lines in the listing): Define the constants `DEFAULT_NAME` and `DEFAULT_AGE` with values 'Unknown' and 21, respectively.

Answer:

```
define( 'DEFAULT_NAME', 'Unknown' );
define( 'DEFAULT_AGE', 21 );
```

β (2 pts., 3 lines in the listing): Assign the value of the form variables to the corresponding program variables.

Answer:

```
$first = $_POST['first'];
$last = $_POST['last'];
$age = $_POST['age'];
```

γ (1.5 pts.): An expression that is true of the value of `$first` is the empty string.

Answer: `empty($first)`

δ (1.5 pts.): An expression that is true of the value of `$last` is the empty string.

Answer: `empty($last)`

ϵ (2 pts.): An expression that is true if `$age` is the empty string or coerces to 0

Answer: `empty($age) or intval($age) == 0`

ζ (1 pt.): The value of `$age` coerced to an integer

Answer: `intval($age)`

η (2 pts.): Output the paragraph with the greeting shown in the screenshot, which includes the values of the variables `$first` and `$last`.

Answer: `echo "<p>Good day, $first $last!</p>";`

θ (5 pts.): The loop that produces the rows in the body of the table

Answer:

```
for ( $i=0; $i<=$age; $i+=5 )
  echo "<tr><td>Years $i-".($i+4 < $age ? $i+4 : $age).
      "</td><td>&nbsp;</td></tr>\n";
```