

Università degli Studi di Camerino
Corso di Laurea in Informatica A.A. 2015/2016
Prova Scritta di Programmazione Web, 22 Marzo 2017 – IX Appello
Docenti: Francesco Tiezzi e Luca Tesei

Nome e Cognome: _____ **Matricola:** _____

Durata della Prova: 45 minuti

Domande a Risposta Aperta: scrivere nello spazio indicato.

1. (6 punti) Spiegare il significato, in modo dettagliato, del codice seguente:

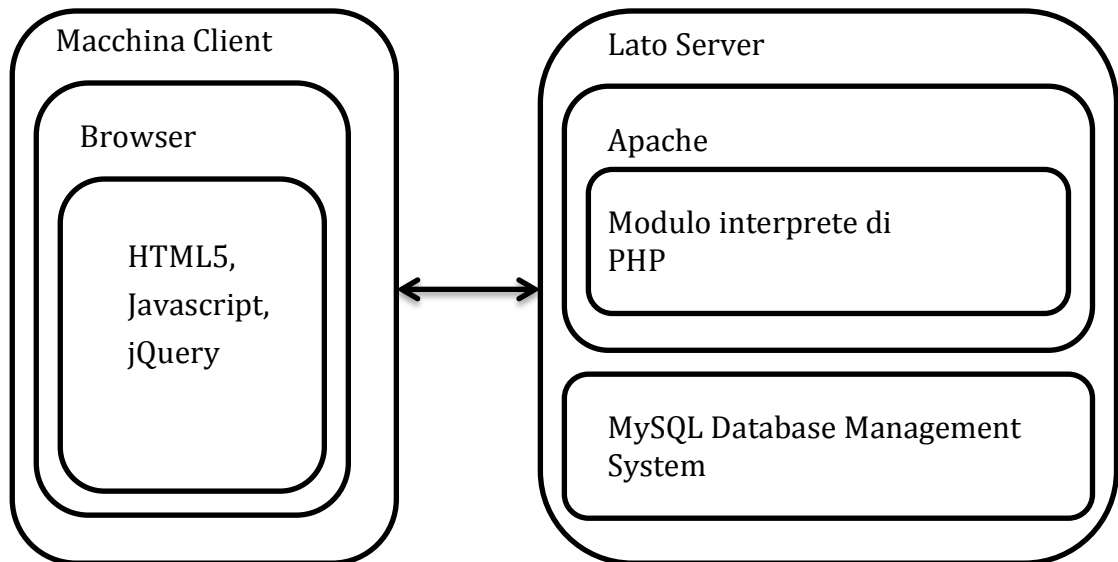
```
...
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html");
    HttpSession session = request.getSession();
    synchronized(session) {
        String heading;
        Integer accessCount = (Integer)session.getAttribute("accessCount");
        if (accessCount == null) {
            accessCount = 0;
            heading = "Welcome, Newcomer";
        } else {
            heading = "Welcome Back";
            accessCount = accessCount + 1;
        }
        session.setAttribute("accessCount", accessCount);
    }
    PrintWriter out = response.getWriter();
    ...
    out.println
        (docType +
         "<HTML>" +
         ... +
         "<H1>" + heading + "</H1>" +
         "<H2> Number of Previous Accesses: " + accessCount + "</H2>" +
         ... +
         "</HTML>");
}
...
```

Si vedano le slides "Core servlets: Session Tracking" e Capitolo 9 del libro di testo.

2. (6 punti) Spiegare in breve il ruolo e il funzionamento della tecnologia JSP, mettendola in relazione con la tecnologia Servlet.

Si vedano le slides "Core servlets: JSP Intro e Overview" e "Core servlets: Invoking Java code with JSP scripting elements", e Capitoli 10 e 11 del libro di testo.

3. (6 punti) Disegnare l'architettura di un classico sito web dinamico collocando nella corretta posizione (e annidando quando possibile) i seguenti componenti/linguaggi: Macchina client, JavaScript, modulo interprete di PHP, Apache, HTML5, jQuery, MySQL Database Management System, Browser, Lato Server



4. (6 punti) Spiegare il significato, in modo dettagliato, del codice seguente contenuto in un file `test.php`:

```

<!DOCTYPE HTML>
<html> <head> <style>.error {color: red;}</style> </head>
<body>
<?php
$nameErr = $name = $email = $website = "";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (empty($_POST["name"])) {
        $nameErr = "Un Nome è obbligatorio";
    } else {
        $name = test_input($_POST["name"]);
    }

    if (!empty($_POST["email"])) {
        $email = test_input($_POST["email"]);
    }

    if (!empty($_POST["website"])) {
        $website = test_input($_POST["website"]);
    }
}

function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}
  
```

```

}
?>

<p><span class="error">* campo obbligatorio.</span></p>
<form method="post" action="<?php echo
htmlspecialchars($_SERVER["PHP_SELF"]);?>">
  Nome: <input type="text" name="name">
  <span class="error">* <?php echo $nameErr;?></span>
  <br><br>
  E-mail: <input type="text" name="email">
  <br><br>
  Sito Web: <input type="text" name="website">
  <br><br>
  <input type="submit" name="submit" value="Invia">
</form>

<?php
echo "<h2>Il tuo input è stato il seguente:</h2>";
echo $name; echo "<br>"; echo $email; echo "<br>"; echo $website; echo "<br>";
?>
</body></html>

```

La pagina web visualizzata in risposta ad una prima richiesta di `test.php` chiede di inserire un nome, una email e un sito web in tre campi di input indicando (in rosso) che il nome è obbligatorio.

Alla pressione del tasto "Invia" i valori inseriti vengono spediti alla stessa pagina con il metodo POST. Il codice php intercetta il fatto che c'è stato un post (if (`$_SERVER["REQUEST_METHOD"] == "POST"`)) e quindi effettua dei controlli sul campo nome scrivendo nella variabile `$nameErr` un messaggio di errore nel caso in cui il campo corrispondente non sia stato compilato (risulta vuoto nell'array globale `$_POST`).

Di seguito, la pagina ripropone lo stesso form di input visualizzando il messaggio di errore (in rosso) accanto ad ogni campo obbligatorio che non è stato compilato precedentemente.

Ancora di seguito la pagina stampa il valore di tutti i valori inviati precedentemente tramite POST, ma privati di spazi precedenti e successivi (funzione `trim`), in cui ogni carattere "quoted" con un backslash è stato "unquoted" (funzione `stripslashes`) e in cui ogni carattere speciale HTML è stato convertito nella corrispondente entità HTML (funzione `htmlspecialchars`).

Domande a Risposta Chiusa: contrassegnare una sola risposta tra le quattro possibili scelte. Ogni domanda vale 1 punto se la risposta è corretta, -0,5 punti se la risposta è errata e 0 punti se non c'è risposta.

1. In HTML, l'elemento **form** consente di:

- a) definire moduli per l'inserimento dati
- b) definire formule matematiche
- c) formattare il testo
- d) creare delle forme geometriche

2. Indicare quale tra le seguenti affermazioni sulle *sessioni* nelle Servlet è **falsa**:

- a) sono gestite tramite un oggetto di tipo **HttpSession**
- b) il metodo **invalidate** dell'oggetto relativo alla sessione elimina l'intera sessione

- c) si imposta il valore di un attributo della sessione tramite il metodo `setStatus` dell'oggetto relativo alla sessione
 - d) si accede all'oggetto per la gestione delle sessioni tramite il metodo `getSession` dell'oggetto relativo alla richiesta
3. Per eseguire una pagina JSP è necessario:
- a) tradurre direttamente la pagina JSP in bytecode
 - b) tradurre direttamente la pagina JSP in linguaggio macchina
 - c) interpretare un comando alla volta della pagina JSP
 - d) tradurre la pagina JSP in Servlet
4. Indicare quale tra i seguenti è l'effetto del comando Javascript `document.getElementById("paragraph").style.color = "red";`
- a) Rende rossi tutti i paragrafi della pagina
 - b) Rende rosso il paragrafo della pagina con attributo `id` uguale a "paragraph"
 - c) Restituisce tutti i paragrafi della pagina che hanno colore rosso
 - d) Resetta lo stile del colore del paragrafo con attributo `id` uguale a "paragraph" se il testo dello stesso è rosso
5. In PHP il risultato della funzione `htmlspecialchars("Testo in grassetto.")` è:
- a) la stringa "Testo in grassetto."
 - b) true
 - c) la stringa "Testo in grassetto ."
 - d) la stringa "Testo in grassetto." in cui la parola "grassetto" è in grassetto
6. In Javascript `XMLHttpRequest` è:
- a) un oggetto che permette di visualizzare del codice XML
 - b) un oggetto che permette di effettuare chiamate AJAX asincrone al server senza ricaricare la pagina
 - c) una funzione che trasforma del codice XML in codice HTML
 - d) un array che contiene le informazioni riguardo la richiesta HTTP della pagina corrente