National University of Computer & Emerging Sciences, Karachi
Fall 2022 (School of Computing)
Midterm II Examination

Course Code: CS-4042	**Course Name:** Information Processing Techniques
Instructors: Mr. Murtaza Munawar Fazal, Mr. Syed Zain Ul Hassan, Mr. Basit Ali, Ms. Abeeha Sattar	
Student ID:	**Section:**
Date: November 01, 2022	**Time:** 11:30 am – 12:30 pm (60 minutes)

Instructions:

- Attempt all questions
- The paper contains 3 questions on 2 pages.
- Return the paper after the exam.

Max Points: 40

Question 1: Multiple Choice Questions **(10 Points)**

I. In a thin client model, what is responsible for the processing of data?

a) Middleware b) Client c) Server d) None of these

II. Which of the following statements are true about JSON?

a) JSON is a data interchange format
b) JSON can be used to transfer data between server and clients
c) JSON does not have any version
d) JSON is a markup language
e) JSON is not a document format

III. Which of the following is correct about SOAP?

a) SOAP is an XML-based protocol for exchanging information between computers.
b) SOAP is a communication protocol.
c) SOAP is for communication between applications
d) All of the above

IV. The WSDL is:

a) Web Servers Definition Language
b) Web Services Developing Language
c) Web Services Definition Language
d) None of these

V. What are the different types of services offered in the cloud?

a) PaaS b) IaaS c) SaaS d) All of the above

VI. If a class inheriting an abstract class does not define all of its functions, then it is known as?

a) Abstract b) A simple class c) Static class d) None of these

VII. How do you represent a Boolean with value "true" in JSON?

a) "true" b) 0 c) True d) 1

VIII. The capability of a system to adapt the increased service load is called

a) Scalability b) Tolerance c) Capacity d) None of these

IX. Which is not a JSON data type?

a) Number b) String c) Array d) Date

X. How do you represent a JSON array of strings?

a) { "days" : { "SUN" , "MON" , "TUE" , "WED" , "THU" , "FRI" , "SAT" } }
b) { "days" = ["SUN" , "MON" , "TUE" , "WED" , "THU" , "FRI" , "SAT"] }
c) { "days" = { "SUN" , "MON" , "TUE" , "WED" , "THU" , "FRI" , "SAT" } }
d) { "days" : ["SUN" , "MON" , "TUE" , "WED" , "THU" , "FRI" , "SAT"] }

Question 2: State the following as True or False and in case of false, justify your answer. **(1 Point each)**

I. Proxy classes for web services can be generated by using the wsdl command in your command prompt. True

II. Distributed objects are always coarse-grained. False: Distributed object architecture is fine-grained

III. One example of a middleware software is CORBA. True

IV. Azure functions are zero maintenance functions. True

V. A web service enables communication among various applications by using open standards such as HTML, XML, WSDL, and SOAP. True

Question 3: Explain the following questions. You may use diagrams where necessary.

I. How is a SOAP message structured? **(3 Points)**

Answer:

```
<Envelope>
  <Header> </Header>
  <Body>
    <Fault></Fault >
  </Body>
</Envelope>
```

II. What benefit does Web Services have over distributed objects? **(4 Points)**

Answer: Distributed object architecture is fine-grained, in that every change to the object has to be propagated across the system. This becomes a disadvantage in terms of network latency. In comparison, web services are coarse-grained and have no disadvantage related to latency.

III. How are Azure functions more efficient in terms of resource requirements and usage? **(4 Points)**

Answer: Azure functions are compute-on-demand and that is scalable. When demand of execution increases, more resources are allocated automatically to the service and when requests fall, all extra resources and application instances drop off automatically.

IV. How do we use Proxy Patterns with collections? Explain and write pseudocode/C# code to demonstrate. **(5 Points)**

Answer:

We can use proxy pattern with collections to create Read-Only Collections or Synchronized collections.

Code would look something like:

```
class proxyList : List {
  List l = new List();
  //just showing for add method
  void add(Object item) {
    return some exception because method is not valid
  }
}
```

V. What would be the equivalent JSON document of the XML document given below? After converting it to JSON document, provide its corresponding C# code. **(4 Points)**

```
<?xml version='1.0'?>
<Library>
  <Department id="1">
    <Bookname>XML Introduction</Bookname>
    <author>O'reilla</author>
    <address>
      <Door>4201 Greenland Avenue</Door>
      <city>New York</city>
      <Country>United States</Country>
    </address>
  </Department>
  <Department id="2">
    <Bookname>Data Warehouse</Bookname>
    <author>Mac meihelline</author>
    <address>
      <Door>5840 texas House</Door>
      <city>Dallas</city>
      <Country> Canada</Country>
    </address>
  </Department>
</Library>
```

Answer:

```
{
  "Library": {
    "Department": [
```

```
    {
     "Bookname": "XML Introduction",
     "author": "O'reilla",
     "address": {
      "Door": "4201 Greenland Avenue",
      "city": "New York",
      "Country": "United States"
     }
    },
    {
     "Bookname": "Data Warehouse",
     "author": "Mac meihelline",
     "address": {
      "Door": "5840 texas House",
      "city": "Dallas",
      "Country": "Canada"
     }
    }
   ]
  }
}

public class Address {
        public string Door { get; set; }
        public string City { get; set; }
        public string Country { get; set; }
}

public class Department {
        public string Bookname { get; set; }
        public string Author { get; set; }
        public Address Address { get; set; }
        public int Id { get; set; }
        public string Text { get; set; }
}
public class Library {
        public List<Department> Department { get; set; }
}
```

VI. Write a ASMX Web Service Function which could encrypt English text using the following encryption scheme: **(5 Points)**

First, the spaces are removed from the text. Let L be the length of this text. Then, characters are written into a grid, whose rows and columns have the following constraints:

$$\lfloor\sqrt{L}\rfloor \leq row \leq column \leq \lfloor\sqrt{L}\rfloor, where\ \lfloor x\rfloor\ is\ floor\ function\ and\ \lceil x\rceil\ is\ ceiling\ function$$

Example

s = "if man was meant to stay on the ground god would have given us roots"

After removing spaces, the string is 54 characters long. $\sqrt{54}$ is between 7 and 8, so it is written in the form of a grid with 7 rows and 8 columns.

ifmanwas
meanttos
tayonthe
groundgo
dwouldha
vegivenu
sroots

- Ensure that rows * columns $\geq L$

- If multiple grids satisfy the above conditions, choose the one with the minimum area, i.e. rows x columns.

The encoded message is obtained by displaying the characters of each column, with a space between column texts. The encoded message for the grid above is:

imtgdvs fearwer mayoogo anouuio ntnnlvt wttddes aohghn sseoau

Function Description

Function accepts the following parameter(s):

- string s: a string to encrypt

Returns

- string: the encrypted string

Constraints

$1 \leq length\ of\ s \leq 81$

s contains characters in the range ascii[a-z] and space, ascii(32).

Sample Input 1		**Sample output 1**	**Explanation**
haveaniceday	=>	hae and via ecy	$L = 12, \sqrt{12}$ is between 3 and 4. Therefore the string is written with 3 rows and 4 columns,
Sample Input 2		**Sample output 2**	**Explanation**
Feedthedog	=>	fto ehg ee dd	$L = 10, \sqrt{10}$ is between 3 and 4. Therefore the string is written with 3 rows and 4 columns.
Sample Input 3		**Sample output 3**	**Explanation**
Chillout	=>	clu hlt io	$L = 8, \sqrt{8}$ is between 2 and 3 but rows * col ≥ length. Therefore the string is written with 3 columns and 3 rows (so we must use 3x3.)

Answer:

```
[WebMethod]
static string encryption(string s)
{
    string result = "";
    s = s.Replace(" ", "");
    var len = s.Length;
    var sqrLen = Math.Sqrt(len);
    int rows = (int)Math.Floor(sqrLen);
    int cols = (sqrLen % 1 == 0) ? (int)sqrLen : (int)sqrLen + 1;

    while (rows * cols < len)
    {
        if (rows < cols)
            rows++;
        else
            cols++;
    }

    for (int i = 0; i < cols; i++)
    {
        for (int j = 0; j < rows; j++)
        {
            int index = i + (j * cols);
            if (s.Count() - 1 >= index)
                result += s[index];
        }
        result += " ";
    }
    return result;
}
```

BEST OF LUCK!