

Information Processing Techniques Lab # 1 & 2

Introduction to Console Applications

Teacher
Murtaza Munawar Fazal



10/25/2019

Practice Questions

- Solutions of each questions should be uploaded on slate by Sunday (October 27, 2019) 11:55 PM
- Submitted code should have all of the files and can be compiled.
- Create One Solution and multiple projects per Lab Activity

Question 1: Write a C# program (Console Application) which will accept string input from user and create a new string with first 4 characters will be in lower case and rest of the letter will be upper case.

Question 2: Write a C# program (Console Application) which will take matrix size as input from user then generate a random matrix and find sum of left diagonals of a matrix.

Question 3: Create a class 'Calculator' which would have the following methods:

- Add
- Subtract
- Multiple
- Divide

Now overload the Add method to accommodate multiple parameters (i.e. 2 parameters, 3 parameters, and n number of parameters).

Question 4: Create a class 'Student' with the following properties:

- Name
- Year of Birth
- Semester
- GPA

Implement the IComparable<> interface so that you can use the Sort method. You should be able to sort the List<Student> in descending order of GPA.

Question 5: You have two character arrays (strings). You have to compare if both the values are equivalent or not using the below method?

```
string string1 = new string(new char[] { 'h', 'e', 'l', 'l', 'o' });
```

```
string string2 = new string(new char[] { 'h', 'e', 'l', 'l', 'o' });
```

```
public static bool CompareObjects (object name, object name2) { }
```

Information Processing Techniques Lab # 1 & 2

Introduction to Threading / Reflections / Web Services (ASMX and WCF)

Teacher
Murtaza Munawar Fazal



10/25/2019

Practice Questions

- Solutions of each questions should be uploaded on slate by Sunday (October 27, 2019) 11:55 PM
- Submitted code should have all of the files and can be compiled.
- Create One Solution and multiple projects per Lab Activity

Question No 1: Generate an array of 1 million random numbers and perform 'searching' on that array to find all occurrences of the given number. Measure time difference on the following techniques:

- Search the Array on one thread (no threading code required)
- Search Array using 5 different threads

You should print the time taken when no thread was used and when 5 different threads are used. Display the indices on which the data was found.

Note: You can use the following code to start a parameterized thread.

```
ParameterizedThreadStart param1 = new ParameterizedThreadStart(MethodToCall);
Thread th1 = new Thread(param1);
th1.Start(new ClassObject(StartingIndex, EndingIndex));
```

Question No 2: Using the multi-threaded code of Question 2, modify the code such that the Search function is accessed by one thread at a time and compare timing with results of Question 2 (i.e. Make the method sequential).

Question No 3: Using C# Reflection, display all the method names that are available in the EntityFramework.dll (File has been uploaded on Slate\IPT\Lab Activity\EntityFramework.dll)

Question No 4: Create an ASMX Web Service for Currency conversion. Your web service will have a fixed list of currencies and their conversion rates from US Dollar hardcoded. You have to provide the following functionalities:

- a. Get List of all Currencies and their rates against 1 US Dollar.
- b. A method to convert currency from one to another. Your method would include 3 parameters, one for Source Currency, one for value to be converted, one for Destination Currency).

Question No 5: Create a Windows Form to consume the web service using a Proxy Class. Generate proxy using the following command:

```
C:\> wsdl <web service url> /out:<output directory>
```

Question No 6: Convert your ASMX Web Service into WCF Web Service and test the results through WCF Test Client.