FROTCOM
INTELLIGENT FLEETS

INSTRUCTIONS

The answers to this sheet should be written in a separate document. Please write down your answers on a Word document, always indicating the number of the question you're referring to.

If the answer requires code, instead of pasting it in the Word document, you can provide a source file or project (zipped without obj or bin folders) for each answer, again, indicating the number of the question (Ex. Question4.cs, Question15.sql or Question4.zip). Code should be written in C# unless stated otherwise. If you provide code through a source file or project, you don't need to write the same code in the document – just choose one.

The task must be completed, and results sent, **within 2 hours**. When finished, please email the results (document and any additional files) to the same address that provided you the task.

Try your best to answer all the questions but don't stress if you find that you don't have enough time – the relevance of this task is in the process, not in the completion. If you don't know the answer, be honest. If you only know half the answer, be honest.

GENERIC / .NET / C#

1. Explain the difference between **overloading** and **overriding**.

2. Explain the difference between an **abstract** and a **virtual** method.

3. With examples, explain what **polymorphism** is.

4. Write a function that receives a string and returns it reversed.
    Ex. "abcd" => "dcba"

5. Write the same function using a different implementation.

6. Write a function that receives a string and returns the words in reverse order.
    Ex. "I want a new smartphone" => "smartphone new a want I"

7. Write the same function using a different implementation.

8. Enumerate three different ways (with examples) on how to create a thread.

9. Implement a simple queue class (thread-safe).

10. Implement a thread-safe class that will read/write to/from a buffer.

HTML / JAVASCRIPT / FRAMEWORKS

11. What are **web workers**?

12. Can a *<section>* contain *<article>* elements? Can an *<article>* contain *<section>* elements? Provide usage examples.

13. Discuss the differences between an HTML specification and a browser's implementation thereof.

14. Briefly describe the correct usage of the following HTML5 semantic elements: *<header>*, *<article>*, *<section>*, *<footer>*.

15. Write the code necessary to create a 300 pixel by 300 pixel *<canvas>*. Within it, paint a blue 100 pixel by 100 pixel square with the top-left corner of the square located 50 pixels from both the top and left edges of the canvas.

16. What is HTML5 Web Storage? Explain **localStorage** and **sessionStorage**.

17. CSS: Name 3 attributes of the position property.

18. CSS: Declare all elements with class of "blue-text" to have a text color of blue.

19. CSS: How would you solve a floated div's parent height?

20. JavaScript: Is JavaScript case sensitive?

21. JavaScript: What is the difference between "==" and "==="?

22. JavaScript: What is the result of "20" + 20?

23. Explain what Bootstrap is.

24. What are the key components of Bootstrap?

25. Explain what is Bootstrap Grid System?

26. How you can add badge to list group in Bootstrap?

27. What CSS class do you use to span 12 columns on medium sized screens but only 6 columns on large screens in the latest version of Bootstrap?

SQL

28. In a basic vehicle tracking system, GPS coordinates, speed and heading, along with vehicle's ignition status (on or off) are recorded periodically. For some vehicles, there are also two additional sensors (temperature and door status), for which, the values are collected every time a GPS position is also collected. There is a database table for GPS information and a second table for additional sensor data. Considering there are thousands of vehicles being tracked, owned by different customers, and thousands of users looking to this information on maps and charts, write the script to create the necessary database tables and the corresponding indexes.
Database server is SQL Server 2016.

29. With the tables previously defined, write down the T-SQL query to get the GPS and sensors data to be shown on a map and chart, accepting the time period and vehicle as parameters.