CS2021 ASSIGNMENT 11 (Due Date: Dec 17, 2021)

Question 1

To make a website to be visible to the public, web pages have to be stored under a web server.

(a) State three common web servers which are commonly installed in a computer to support this service.

(b) State the full name of HTML.

(c) The syntax of HTML is embraced under an application protocol. What is the full name for this protocol?

(d) If there is any command with syntax error in a HTML file, what will the interpreter in a browser will do? The browser will not render the HTML file or the browser will render the web page but ignoring the erroneous commands.

Answer:

(a) Microsoft Internet Information Services (IIS) server, Apache, NGINX.

(b) Hypertext markup language.

(c) Hypertext transfer protocol.

(d) The browser still renders the web page but the erroneous commands are ignored.

Question 2

(a) What is the effect of

```
<i>John Sum</i>
```

in the display of John Sum in a web page?

(b) What is the effect of

```
<b>John Sum</b>
```

in the display of John Sum in a web page?

(c) What is the difference in the usage of the tags $< ol >$ and $< ul >$ in listing of items?

(d) What is the purpose of the tagging $< p >$ and $< /p >$ in an HTML file?

(e) HTML is a compiled language or an interpreted language?

Answer:

(a) The string "John Sum" will be formatted in italic.

(b) The string "John Sum" will be formatted in bold face.

(c) The tag $< ol >$ is used for ordered listing. The item being listed will be given with a number. The tag $< ul >$ is used for unordered listing.

(d) The tag $< p >$ (and $< /p >$) is used for indicating that the texts inside the p-tag is a paragraph. So, (i) the texts will be displayed with a small vertical space between the paragraph and the last line of text. Moreover, (ii) the first character of the paragraph will be displayed on the leftmost position.

(e) HTML is an interpreted language.

Question 3

(a) C language is a compiled language or an interpreted language?

(b) In the heading of a C program, it always will include a number of header files like below.

```
#include <stdio.h>
#include <math.h>
```

What is the purpose for adding such header files?

(c) What is the purpose for the declaration of the variables and their data types in the beginning of the program?

(d) State three C compilers which are commonly installed in a computer.

(e) Owing to let a compiler to convert the source code of "printf" to machine code, which header file has to be included?

(f) If we would like a C compiler to ignore a command, what could we add in the program source code file?

Answer:

(a) Compiled language.

(b) Some functions, like *printf()* and *scanf()*, their subroutines are included in the header file "stdio.h". The header files are used for a compiler to get the codes for the functions.

(c) First, it lets the compiler to check if there is data type mismatch in a command. Second, it lets an operating system to allocate the working space for running the program successfully.

(d) For Windows operating systems, two exemplar compilers are DevC and Borland Turbo C compilers. C IDE is a compiler for MacOS. GCC is a compiler for either Linux or Unix operating system.

(e) stdio.h

(f) Add remark (or comment) in the way like the following.

```
/*
    for(i=0; i<5; i++)
        printf("*");
*/
```

The for-loop will be ignored during compilation.

Question 4

(a) What is the difference between a pseudo-code program and a C program source code?

(b) A C program which can be compiled by a C compiler X. It might not be compiled by another C compiler Y. Explain the reason(s) why?

Answer:

(a) A pseudo-code is a summary of the procedure to be executed in the program. It could be a list of steps written in English or a list of steps written in programming language. A C program source code is the detail listing of the commands which have to be conformed to the syntax of the programming language.

(b) Different compilers might have slightly different design for the set of header files. For instance, the subroutine a function call, say xxx() might have been included in the abc.h of the X compiler. But, the subroutine of xxx() might have not been defined in the abc.h in the Y compiler. Thus, the source code which can be compiled by the X compiler might not be able to be compiled by the Y compiler.