**THEORY TASK #1 THURSDAY 8/APRIL/2021**

***- Unicode is a 4Byte schema, i.e. it use combination of 32 bits for single character storage.***

***- Java is based on Unicode schema.***

***- But Java Char datatype occupy 2Byte, Byte datatype occupies 1 Byte, Short datatype occupies 2Byte.***

The encodings for languages with large character sets have variable lengths. Some common characters are encoded as single bytes, other require two or more byte. To solve these problems, a new language standard was developed i.e. Unicode System.

**UNICODE:**

Unicode is an **Information Technology Standard** used for encoding, representation, and handling of text expressed in writing system. Unicode can be implemented by different character encodings. The Unicode standard defines Unicode Transformation Formats (UTF) **UTF-8, UTF-16,** and **UTF-32**, and several other encodings.

**UTF-8:-**

UTF-8 is the dominant encoding for mostly on Unix OS, which uses one byte for 128 characters (ASCII) and goes up to 4 bytes for other characters.

**UTF-16:-**

UTF-16 is the 16-bit Unicode system is capable of encoding all the code points of Unicode. The first Unicode format was designed using 16 bits because the primary machines at that time were mostly 16-bit PCs. At that time, when Java language was created, then Unicode standard was accepted and used in Java.

So, it is concluded that the data types has the memory that is used in UTF-16. And hence, the “char” data type in Java has the size of 2 bytes but not 4 bytes, as the size of char data type is used as in UTF-16. And Byte data type occupies one byte because it is ASCII code points.

**THEORY TASK #2 SATURDAY 10/APRIL/2021**

**Q1:** Research about data types, in attached snip, when we divide **12(int)/24(int)**, and assigning the answer in float variable **d,**but when be console out, it prints 0.00, when expected answer is 0.5.

**Answer:** When we divide two integers the result will also be integer, So as in that sense if we divide **12(int)/24(int)** the result will be in integer but as we know integer data type uses arithmetic division so the answer will be rounded off to **0**. But, as we see the code, the result is assigned as float data type so, the result of division will be converted into the float, and hence, the answer we get is **0.0**.