**Text Representation** standard are needed as text and documents can be shared and viewed on many different computer systems that are being used each day.

**C2: The purpose of implications of using codes to represent character sets.**

* A byte is made up of 8 bits, which can be arranged in 256 different combinations.
* There are a common set of codes for the alphabet and numbers, which take up **62** (0-9, A-Z, a-z) of the 256 possible codes in a byte.
* The 8-bit code for a character can be **represented in denary or hexadecimal** to give a sequence:
1. For example: The letter A is stored in memory as 0 1 0 0 0 0 0 1. It can be represented as 65 in denary or 41 in hexadecimal.
* The difference between **upper and lower case** characters is **bit 6**. (The 6th bit of the byte)
* If the 6th bit is ‘1’, then the letter is lower case.
* If the 6th bit is ‘0’, then the letter is upper case.
1. **‘A’ =** 0 1 **0** 0 0 0 0 1
2. ‘**a**’ = 0 1 **1** 0 0 0 0 1 (the number in bold is the 6th bit)
* The purpose of this structure makes it easy for computer systems needing a user ID or word processors to recognise upper or lower case versions of a character by ignoring bit 6.
* Conversion between upper and lower case is also easier by setting bit 6 to 0 or 1.
* This adds possibilities for the font used for a character set as the appearance of numbers and alphabetic characters is easily changed using a different font.
* A font can also make use of the other codes for shapes or accented characters

**C2: The features and uses of common character sets:**

**ASCII: (**American Standard Code for Information Interchange / ISO-IR-006)

* Developed in 1960 (October 6th)
* This character set is used for the English Language.
* It is based on up to 7 bits with 128 different characters, which include control characters:
1. Carriage Return (CR)
2. ESC (Escape)
* Extended ASCII uses 8 bits.

**UNICODE:**

* Developed in 1987.
* This is a character set that has been used by Windows and most websites since 1990.
* It uses between 1 and 4 bytes for each character. This gives over a million possible characters, which makes it a good system for using multiple languages.
* It is compatible with ASCII because they share the same codes for numbers and the alphabet.