**Presentation Notes**

1. What does the ASCII acronym stand for?

**American Standard Code for Information Interchange**

1. What is the ASCII code used for?

**ASCII is used for representing & storing text in computers. As computers can only understand binary numbers so the text symbols must be encoded as numbers.**

1. Encoding characters (i.e. letters on the keyboard) into ASCII code numbers  
   1. What is the ASCII code for the letter “A”  
      **65.**
   2. What is the ASCII code for the letter “a”  
      **97.**
   3. Why are they different?  
      **Lowercase & uppercase symbols are different so they must be registered differently.**
   4. What is the ASCII code for the space bar?

**The ASCII code for the space bar is 32.**

1. Decoding ASCII code numbers into characters and letters   
   1. What character corresponds to ASCII code 61 decimal  
      **=**
   2. What character corresponds to ASCII code 8 decimal  
      **Backspace**
   3. Why is the character 8 not the same as ASCII code 8  
      **Character 8 is code 56. That is because the symbols up to 47 are punctuation and miscellaneous.**
   4. What is the range of non-printable characters in ASCII

**0-31**

1. How would you code the string “Hello” in ASCII?  
   **72,101,108,108,111**
2. How would you code the string “127” in ASCII?  
   **49,50,55**
3. What is the difference between 127 and “127”?

**127 would be the decimal code, whilst “127” is the character.**

**Sudent Questions**

1. Why do computers have to convert characters (i.e. letters on the keyboard) into numbers? Why can’t computers just use the letters directly?  
   **Computers have to convert characters into numbers as computers do not have the same understanding as humans do. So the computers must convert the characters into numbers.**
2. How do computers communicate with people who speak different languages and use different alphabets? What is used instead of the ASCII code table?

**Unicode is used for non-Latin-based languages.**

1. Research online-documentation for the Python **ord()** function. Provide some sample code that demonstrates the use of the **ord()** function.

**Ord returns an integer representing Unicode code point of the given Unicode character.**

**Example: print(ord(‘a’))**

1. Research online-documentation for the Python **chr()** function. Provide some sample code that demonstrates the use of the **chr()** function.

**Chr returns a character (string) whose Unicode code point is the integer**

**Example: print(chr(97))**

1. Write a Python program that uses the ord() and chr() functions to do the following:
   1. Read a single character (i.e. single letter or keyboard symbol) from the console input.
   2. Convert the character to an ASCII code number.
   3. Add 3 to the code number.
   4. Convert the new code number back to a character (i.e. single letter or keyboard symbol)
   5. Print the new character to the console output.

chr1 = (input("Type a single character:"))

ord1 = ord(chr1)

ord2 = ord1 + 3

chr2 = chr(ord2)

print("The character added by 3 is:", chr2)

1. Enhance your program to add the following features:
   1. After reading the single character from console input, check to make sure that the character is a letter (i.e. a to z or A to Z). Print a warning message if the character is not a letter.
   2. After converting the code number back to a character, print a “\*” if the character is not a letter.

chr1 = (input("Type a single character:"))

ord1 = ord(chr1)

isChrOk = False

if(ord1 >= ord('a') and ord1 <= ord('z')) :

isChr0k = True

elif (ord1 >= ord('A') and ord1 <= ord('Z')) :

isChr0k = True

else:

print("Warning: Typed character is not a letter")

ord2 = ord1 + 3

if (ord2 >= ord('a') and ord2 <= ord('z')) :

chr2 = chr(ord2)

elif (ord2 >= ord('A') and ord2 <= ord('Z')) :

chr2 = chr(ord2)

else:

chr2 = '\*'

print("The character added by 3 is:", chr2)

**Extension (Optional)**

1. Extend your program to operate on a string read in from the console input.
   1. Use a loop to process the string as a sequence of single characters
   2. Use your original code process the characters
   3. Append the characters to make a new output string
   4. Print the new string to console output