**Presentation Notes**

1. What does the ASCII acronym stand for?

ASCII stands for American Standard Code for Information Interchange.

1. What is the ASCII code used for?

The ASCII code is used for representing and storing text in computers and sharing information with other computers in the binary, the only language computers understand.

1. Encoding characters (i.e. letters on the keyboard) into ASCII code numbers  
   1. What is the ASCII code for the letter “A”
      1. 65
   2. What is the ASCII code for the letter “a”
      1. 97
   3. Why are they different?
      1. Since both characters are different types of characters even though they are both a, to represent any one of these characters, there must be a different binary number for each character.
   4. What is the ASCII code for the space bar?
      1. 32
2. Decoding ASCII code numbers into characters and letters   
   1. What character corresponds to ASCII code 61 decimal
      1. The sign is =.
   2. What character corresponds to ASCII code 8 decimal
      1. Backspace.
   3. Why is the character 8 not the same as ASCII code 8
      1. Both are different since we are regarding the character 8, which is different than the representation of the decimal 8.
   4. What is the range of non-printable characters in ASCII
      1. 0 - 31

1. How would you code the string “Hello” in ASCII?
   1. The code is written below:
      1. 34 72 101 108 108 111 34
2. How would you code the string “127” in ASCII?
   1. 34 49 50 55 34
3. What is the difference between 127 and “127”?
   1. The decimal value of 127 is set as a reference point on the ASCII table, counteractively; the “127” is set as a string value on the ASCII table.

**Student 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?
   1. Since computer voltage is either on, or off, the computer system is unable to define every number in the decimal system, therefore the binary system is in place for 1s and 0s for every circuit that is on and off.
2. How do computers communicate with people who speak different languages and use different alphabets? What is used instead of the ASCII code table?
   1. Other languages are ISO 646, YUSCII, and altered versions of ASCII according to one nation’s demands.
3. Research online-documentation for the Python **ord()** function. Provide some sample code that demonstrates the use of the **ord()** function.
   1. The ord() function returns the integer value of the character in the Unicode set of the ASCII table.
   2. print(ord(‘a’))
4. Research online-documentation for the Python **chr()** function. Provide some sample code that demonstrates the use of the **chr()** function.
   1. The chr() function is used to input a integer for a return of the string value of that integer in the Unicode ASCII table set.
   2. print(chr(97))
5. 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.
      1. print(chr(55))
   2. Convert the character to an ASCII code number.
      1. print(ord(‘a’))
   3. Add 3 to the code number.
      1. a = ord(‘a’)
      2. print(a + 3)
   4. Convert the new code number back to a character (i.e. single letter or keyboard symbol)
      1. a = ord(‘a’)
      2. b = a + 3
      3. print(chr(b))
   5. Print the new character to the console output.
      1. a = ord(‘a’)
      2. b = a + 3
      3. print(chr(b))
6. 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.

a = input('Enter a letter.\n')

x = (a.isalpha())

if x is False:

print('Your inputed value was not a letter - \*')

else:

a = ord(a)

if a <= 122 and a >= 65:

b = a + 3

(chr(b))

print(chr(b), 'is your new character.')

**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