**Presentation Notes**

1. What does the ASCII acronym stand for?

* American Standard Code For Information Exchange

1. What is the ASCII code used for?

* Way of changing symbols to numbers
* Representing and storing text in 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
  1. What is the ASCII code for the letter “a”
* 97
  1. Why are they different?
* Upper case and lower case are different symbols. The computer doesn't really know what the alphabet is or how to read and write.  
  1. What is the ASCII code for the space bar?
* 32

1. Decoding ASCII code numbers into characters and letters   
   1. What character corresponds to ASCII code 61 decimal

* =
  1. What character corresponds to ASCII code 8 decimal
* Backspace
  1. Why is the character 8 not the same as ASCII code 8
* Character "8" is text symbol, code 8 is an number. Symbols and numbers are different things to a computer.  
  1. What is the range of non-printable characters in ASCII
* Codes 0 to 31

1. How would you code the string “Hello” in ASCII?

* H e l l o

72 101 108 108 111

1. How would you code the string “127” in ASCII?

* 1 2 7

49 50 55

1. What is the difference between 127 and “127”?

* 127 is an integer number. Computers don't need to use ASCII for numbers.  
  "127" is a string of text symbols. A human might see this as the number 127. A computer doesn't know it's a number.

**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?

* to keep calculations simple and convert into binary online, computers use the binary number system.

1. How do computers communicate with people who speak different languages and use different alphabets? What is used instead of the ASCII code table?

* unicode virtually eliminates this problem as all the character code points were standardized. Another major advantage of Unicode is that at its maximum it can accommodate a huge number of characters. ... ASCII uses an 8-bit encoding while Unicode uses a variable bit encoding. 2.Unicode is standardized while ASCII isn't.

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

* The ord() function in Python accepts a string of length 1 as an argument and returns the unicode code point representation of the passed argument. For example ord('B') returns 66 which is a unicode code point value of character 'B'.

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

* The chr() method returns a string representing a character whose Unicode code point is an integer. ... The chr() method returns a character whose unicode point is num, an integer. If an integer is passed that is outside the range then the method returns a ValueError.

y = chr(65)

print(y)

A

print(chr(123))

{

 print(chr(36))

$

x = ord('A')

print(x)

print(ord('ć'))

print(ord('ç'))

print(ord('$'))

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.

chrIn = int(input("Please type in a single character:"))

ordIn = ord(chrIn)

ordOut = ordIn + 3

chrOut = chr(ordOut)

print("The character shifted by 3 is:", chrOut)

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.

**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