**Presentation Notes**

1. What does the ASCII acronym stand for?

American Standard Code for Information Interchange

1. What is the ASCII code used for?

* Representing and storing text in computers
  + Computers can only understand numbers (binary)
  + 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?  
      Upper case and lower case are different symbols. The computer doesn't really know what the alphabet is or how to read and write.
   4. What is the ASCII code for the space bar?

32

The ASCII code also includes some "un-printable" characters.

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 text symbol, code 8 is an number. Symbols and numbers are different things to a computer.
   4. 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?
   1. Computers can only understand numbers (binary) and Computers can only understand numbers (binary)
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. They convert the ASCII numbers into characters that humans can understand.
   2. The characters are within UTF and Unicode
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 in Python accepts a string of length 1 as an argument and returns the unicode code point representation of the passed argument.
   2. Input: a
   3. Output: 97
4. Research online-documentation for the Python **chr()** function. Provide some sample code that demonstrates the use of the **chr()** function.
   1. The Python chr() function is a built-in Python function that returns the string representing a character whose Unicode is an integer.
   2. >>> print(chr(98))
   3. b
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.
   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.

singleCharacter = input ("Enter a single character: ")

print ("Your number is", (ord(singleCharacter)))

newNumber = (ord(singleCharacter) + 3)

print ("The new number is",newNumber)

print ("The new character is",(chr(newNumber)))

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.

singleCharacter = input ("Enter a single character: ")

if ((ord(singleCharacter)) < 65 or (ord(singleCharacter)) > 122) :

print ("Warning: Your character is invalid")

print ("Your number is", (ord(singleCharacter)))

newNumber = (ord(singleCharacter) + 3)

print ("The new number is",newNumber)

print ("The new character is",(chr(newNumber)))

if (newNumber < 65 or newNumber > 122) :

print ("\*")

**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

I tried doing it and this is the best I could do.

singleCharacter = str(input("Enter characters: "))

indexNumber = 0

#while (indexNumber < 100) :

while singleCharacter != "":

if indexNumber == "" :

break

print (ord(singleCharacter[indexNumber]))

indexNumber = indexNumber + 1

stringOutput = chr(singleCharacter)

print ("Your string is", stringOutput)