**Presentation Notes**

1. What does the ASCII acronym stand for?

This stands for American Standard Code for information interchange

2. What is the ASCII code used for?

It is used for representing and storing text in computers

·         Computers can only understand numbers(binary)

·         Text symbols must be encoded as numbers

3. Encoding characters (i.e. letters on the keyboard) into ASCII code numbers

a. What is the ASCII code for the letter “A”

65

b. What is the ASCII code for the letter “a”

97

c. 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.

d. What is the ASCII code for the space bar?

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

4. Decoding ASCII code numbers into characters and letters

a. What character corresponds to ASCII code 61 decimal

The = symbol

b. What character corresponds to ASCII code 8 decimal

backspace

c. Why is the character 8 not the same as ASCII code 8

it is because it is a number and not a symbol

d. What is the range of non-printable characters in ASCII

The range is from 0-32

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

You would type 72,101,108,108,111

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

You would type 49, 50, 55

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

The difference is that 127 is the integer number and “127” is a text symbol

**Student Questions**

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 convert letters to numbers because the computer does not understand letters. Rather it understands numbers. The computer uses the binary system because there are fewer computational errors.

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

 They have different tables that add characters to the ASCII table that the language might need. Some of the other tables used are 7-bit codes, 8-bit codes, and Unicode.

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

 The ord() function is used to classify what letter represents the integer. For example, if I type print((ord(‘a’)), I will be displayed by the answer of 97 in decimal. Another example is if I type ord(J), I will be displayed with an answer of 74 in decimal

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

 The chr() function does the opposite of ord(). If you enter this code: print((chr(99)), I will get displayed with the answer of c. Another example is print((chr(74)), I will displayed with the answer of 72 in decimal.

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.

num = ord(input(""))

print(num)

num = ord(input(""))

num = num + 3

print(num)

num = ord(input(""))

num = num + 3

print(num)

print(chr(num))

num = ord(input(""))

num = num + 3

print(num)

print(chr(num))

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.

num = ord(input(""))

print(num)

print(chr(num))

if (num < 57):

 if (num > 47):

  print ("Warning")

num = ord(input(""))

print(num)

if (num < 57):

 if (num > 47):

  print ("\*")

**Extension (Optional)**

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