**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, and encoding text for electronic communication

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 and the computer doesn’t know the alphabet so it looks at them as individual symbols.

* 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 a text symbol while code 8 is a number. Symbols and numbers are different to a computer.

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

Codes 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 is an integer number while “127” is a string of text.

**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?

Because it is too complex for the computer and it can’t understand the English language.

1. 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.

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

The ord() function takes a character and returns an integer.

# inbuilt function return an

# integer representing the Unicode code

value = ord("A")

# writing in ' ' gives the same result

value1 = ord('A')

# prints the unicode value

print value, value1

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 character (a string) from an integer.

print(chr(97))

print(chr(65))

print(chr(1200))

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.

num1 = ord(input("Type a character: "))

num2 = num1 + 3

num3 = chr(num2)

print(num3)

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.

num1 = ord(input("Type a character: "))

if (num1 <= 122 and num1 >= 97) or (num1 <= 90 and num1 >= 65):

num2 = num1 + 3

num3 = chr(num2)

else:

print("the character has to be a letter")

if (num2 <= 64) or (num2 >= 91 and num2 <= 96) or (num2 >= 123):

print(num3, "\*")

else:

print(num3)

**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