**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?

The computer does not comprehend human language. It only understands binary numbers. You have to convert characters

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

In order to communicate with humans a computer will convert the ASCII language to letters, numbers, symbols and punctuation marks so that humans can understand what the computer is to communicate

3. 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 repersentation of the passed argument. For

example, ord(‘B’) returns 66 which is a unicode code point value of character

‘B’.

# inbuilt function return an

# interger representing the Unicode code

Value= ord(“A”)

# writingin ‘ ‘ gives the same result

Value1= ord(‘A’)

# prints the Unicode value

4. Research

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

demonstrates the use of the **chr()** function.

 a. As you can see above3e,  the chr() function takes a single parameter and returns the corresponding character of the integer ASCII value

b. >>>print(chr(98))

B

>>>print(chr(555))

ȫ

5. Write a Python program that uses the ord() and chr() functions to do the following:

a. Read a single character (i.e. single letter or keyboard symbol) from the console input.

b. Convert the character to an ASCII code number.

c. Add 3 to the code number.

d. Convert the new code number back to a character  (i.e. single letter or keyboard symbol)

e. Print the new character to the console output.

myCharacter = input ("Please enter a character ")

print (ord(myCharacter))

myCode = (ord(myCharacter))

print (myCode + 3)

print ("Your new character is:")

print(chr(myCode + 3))

6. Enhance your program to add the following features:

a. 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.

b. After converting the code number back to a character, print a “\*” if the character is not a letter.

myCharacter = input ("Please enter a character ")

print (ord(myCharacter))

myCode = (ord(myCharacter))

print (myCode + 3)

print ("Your new character is:")

print(chr(myCode + 3))

myNewCode = (myCode + 3)

if (myNewCode > 127 or myNewCode < 65):

 print ("\*")

Extension (Optional)

7. Extend your program to operate on a string read in from the console input.

a. Use a loop to process the string as a sequence of single characters

b. Use your original code process the characters

c. Append the characters to make a new output string

d. Print the new string to console output

myCharacter = input ("Please enter a character ")

index = 0

while (index < 4):

 print(ord(myCharacter[index]))

 index = index + 1