**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

1. **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

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’.

# inbuilt function return an

# interger representing the Unicode code

Value= ord(“A”)

# writingin ‘ ‘ gives the same result

Value1= ord(‘A’)

# prints the Unicode value

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

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

>>>print(chr(98))

B

>>>print(chr(555))

ȫ

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.**

myCharacter = input ("Please enter a character ")

print (ord(myCharacter))

myCode = (ord(myCharacter))

print (myCode + 3)

print ("Your new character is:")

print(chr(myCode + 3))

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.**

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)**

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

myCharacter = input ("Please enter a character ")

index = 0

while (index < 4):

print(ord(myCharacter[index]))

index = index + 1