Lesson 11: ASCII and More on *char*

Things you can’t do:

Character type *char* and *String* types can’t be stored into each other. The following lines of code are **illegal**:

```
char ch = aString;      //where aString is a String: illegal
char ch = “A”;          //illegal
String x = xChar;       //where xChar is a char: illegal
String x = ‘X’;         //illegal
```

Surprisingly legal:

Strangely enough the following **is** legal:

```
int x =1;
char ch = ‘A’;          //ASCII code for ‘A’ is 65… (more on ASCII below)
int y = x + ch;                 //This is legal!
System.out.println(y);          //66

int z = ch;                     //This is legal!
```

Illegal!

Storing an *int* type into a *char* is illegal.

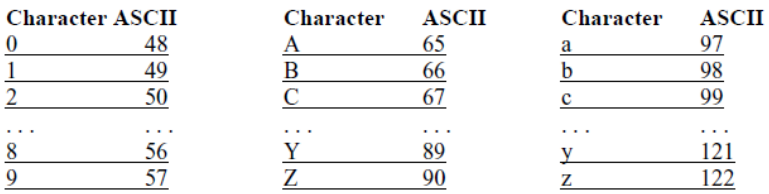
```
char ch = j;            //Illegal…assuming j is an int
```

Why is this illegal? It’s because *char* can take on Unicode values from 0 – 65536 (two bytes)while *int* types can go over 2 billion. The compiler justly complains about “possible loss of precision” and refuses to do it. Use casting as a way around this.

```
char ch = (char)j;      //Legal…assuming j is an int and less than 65,536
```

ASCII (pronounced “ask-key”) codes:

Why does the code in middle section above work? It’s because characters are just numbers. For example, capital A is stored as a 65. That’s why we got 66 above. All characters (letters, numbers, symbols, etc) are stored as numbers. Some ASCII codes that you **should know** are:



Character	ASCII	Character	ASCII	Character	ASCII
0	48	A	65	a	97
1	49	B	66	b	98
2	50	C	67	c	99
. . .	. . .	. . .	. . .	. . .	. . .
8	56	Y	89	y	121
9	57	Z	90	z	122

For more on ASCII codes, see Appendix D.

Conversion between *String*s and characters:

Let’s look back at the top section of this page. What do you do if you absolutely have to convert a *String* into a character or vice versa?

a. **Conversion of a *String* into a character**

```
String s = “W”;
char a = s.charAt(0); //a now equals ‘W’
```

b. Conversion of a character into a *String*

```
char a = ‘X’;
String s = “” + a;        //concatenation of a string and a character is permit-
                          //ed. The result is a String. The trick is to make the
                          //String we are concatenating an empty String (“”).
```

Conversion from capital to small:

A way to convert capital-letter characters into small-letter characters is to add 32. Look in the chart above…capital A is 65……small a is 97…….a difference of 32.

```
char bigLetter = ‘H’;
char smallLetter = (char)(bigLetter + 32); //(bigLetter + 32) is an int that must be
                                           //cast…see # 3 on previous page.
System.out.println(smallLetter); //h
```

What are you? (just ask)

We can ask the following questions of a character (answers are always *true* or *false*),

c. “are you a digit?”

```
char ch = ‘a’;
System.out.println( Character.isDigit(ch) ); //false
char ch = ‘3’;
System.out.println( Character.isDigit(ch) ); //true
```

d. “are you a letter?”

```
char ch = ‘a’;
System.out.println( Character.isLetter(ch) ); //true
char ch = ‘3’;
System.out.println(Character.isLetter(ch) ); //false
```

e. “are you a letter or a digit?”

```
char ch = ‘a’;
System.out.println( Character.isLetterOrDigit(ch) ); //true
char ch = ‘3’;
System.out.println( Character.isLetterOrDigit(ch) ); //true
```

f. “are you whitespace?”….(new line character, space and tabs are whitespace)

```
char ch = ‘ ’;
```

```
System.out.println( Character.isWhitespace(ch) ); //true
char ch = ‘p’;
System.out.println( Character.isWhitespace(ch) ); //false
```

g. “are you lowercase?”

```
char ch = ‘a’;
System.out.println( Character.isLowerCase(ch) ); //true
char ch = ‘A’;
System.out.println(Character.isLowerCase(ch) ); //false
```

h. “are you uppercase?”

```
char ch = ‘a’;
System.out.println( Character.isUpperCase(ch) ); //false
char ch = ‘A’;
System.out.println( Character.isUpperCase(ch) ); //true
```

Conversion to upper case:

We can convert a character to upper case as follows:

```
char ch = ‘d’;
char nn = Character.toUpperCase(ch);
System.out.println(nn); //D
```

Conversion to lower case:

We can convert a character to lower case as follows:

```
char ch = ‘F’;
char nn = Character.toLowerCase(ch);
System.out.println(nn); //f
```