

Topic: Character Class Ref: CSC15 PAL WS Text Manipulation

A Java library is just a collection of classes that have been written by somebody else already. You can use those classes in your code. This lets you expand what Java can do and rely on code that other people have tested instead of doing everything yourself. Some of these are automatically imported (like String, Integer, Character, Math, Exception) and you can include others like Random, Scanner, File, Exception using the import directive.

Character Class

The Character class wraps a value of the primitive type char in an object. An object of class Character contains a single field whose type is char. You can also read about the wrapper Integer class in the book.

The Character class provides a large number of static methods for determining a character's category (lowercase letter, digit, etc.) and for converting characters from uppercase to lowercase and vice versa.

There is a section about type char in chapter 4 of the textbook. A list of useful Character class methods is given in Appendix A of your book.

One of the most important ideas is that the values of type char have corresponding integer values. There is a character with value 0, a character with value 1, a character with value 2 and so on. You can compare different values of type char using less-than and greater-than tests, as in:

```
if (ch >= 'a') { .. }
```

All of the lowercase letters appear grouped together in type char ('a' is followed by 'b' followed by 'c', and so on) and all of the uppercase letters appear grouped together in type char ('A' followed by 'B' followed by 'C' and so on). Because of this, you can compute a letter's displacement (or distance) from the letter 'a' with an expression like the following (this expression assumes the variable letter is of type char and stores a lowercase letter):

```
letter - 'a'
```

Do you think this number can be used to index an array?

Going in the other direction, if you know a character's integer equivalent, you can cast the result to char to get the character. For example, suppose that you

want to get the letter that is 8 away from 'a'. You could say:

```
char result = (char) ('a' + 8);
```

This assigns the variable result the value 'i'.

As in these examples, you should write your code in terms of displacement from a fixed letter like 'a' rather than including the specific integer value of a character like 'a'.

You probably want to look at the String and Character classes for other useful methods (e.g., there is a toLowerCase method in each).

The String methods are mostly instance methods because Strings are objects. The Character methods are all static because char is a primitive type. For example, assuming you have a variable called s that is a String, you can turn it to lowercase by saying:

```
s = s.toLowerCase();
```

This is a call on an instance method where you put the name of the object first. But char values are not objects and the toLowerCase method in the Character class is a static method. So assuming you have a variable called ch that is of type char, you'd turn it to lowercase by saying:

```
ch = Character.toLowerCase(ch);
```

Ex. 1. Write a method that takes a character ch as a parameter and returns the integer value of ch. What is the integer equivalent of the letter 'a'? Is it different from 'A'?

Ex.2. Using the above information, write a method that inputs a character returns the distance from 'a'. First determine if the incoming character is part of the alphabet. If it's not then returns a signal value like -1. Convert the incoming character to lowercase before calculating distance

Ex.3. String to array of chars

Ex.4. Write a program that reads in a string called word and reports the letter frequency. So if you input "aBAca" it should return 311 since there were 3 a's 1 b and 1 c. You would need to create an array of integers and set the "distance" from 'a' information to calculate the index for each corresponding letter of the alphabet. This array can be used to calculate the number of times a letter occurs. Do you see how the frequency of 'a' can be stored in position 0 of this array?

Use these methods for converting a char array to string and vice versa

```
char[] charArray = str.toCharArray(); // create a char Array from a string  
  
String s = String.valueOf(charArray); // convert to string
```

