Chars and Strings

For hw01 you need to read some code that uses characters and strings. In class we did not get to discuss all the facts about characters and strings. Therefore, this handout will explain the main ideas. The textbook has more details.

Types of Data

A computer stores and processes data. There are a few different types of data. These are the first types we shall look at:

<table>
<thead>
<tr>
<th>Type</th>
<th>Stores</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>whole numbers</td>
</tr>
<tr>
<td>double</td>
<td>numbers with fractional parts</td>
</tr>
<tr>
<td>char</td>
<td>keys on the keyboard</td>
</tr>
<tr>
<td>string</td>
<td>sequence of chars</td>
</tr>
</tbody>
</table>

Creating and Assigning Variables

You store values in little storage boxes in the computer. To request a little storage box, you type a variable definition by putting the type of the variable followed by one or more variable names. For example:

```c
int high_temp, days_in_may;
double body_temp, half;
char my_initial;
string CS_building;
```

You can store values in these boxes by using the assignment operator which is written as = and is pronounced gets. Here are some examples:

```c
high_temp = 78;
days_in_may = 31;
body_temp = 98.6;
half = 0.5;
my_initial = 'B';
CS_building = "Halligan";
```

The rule is simple: the name of the variable goes on the left, the = goes in the middle, and the value you are storing goes on the right. Note: The variable name goes on the left side.

Look at the sample values shown. An "int" variable can only store whole numbers, and we store the integer value 78. A "double" variable can store numbers with fractional parts, and the value we store in the first example has a fractional part of .6, and the value we store in the second example has a value of 0.5.

A "char" value can store any single key on the keyboard. The notation for a char value uses the single quotation marks to surround the character. In this case 'B' represents the single letter B on the keyboard. A char can be any digit, any piece of punctuation, a space, even the Enter key and the Tab key. For the last two, we use the notations '
' and '\t'.

Finally, a string is any sequence of characters. We surround the string value with regular quotation marks. Any characters may appear in the string including the special notation for Enter and Tab (\n and \t).

Important Fact about Char Variables

Computers can only store numbers. When you write something like

```c
char first_let;
first_let = 'A';
```

the computer stores the letter A as a number. Every key on the keyboard has a numerical code that computers use to represent the characters. The letter 'A' is stored as the number 65. The letter 'B' is stored as the number 66. The code that assigns numbers to letter is called ASCII. To see the complete list, you can type, in your Linux terminal window,

```c
man ascii
```

and look in the column marked "dec" (for decimal). To see more of the table, press the Enter key. To stop viewing the table, press the 'q' key.

Operating on Values

Computers store values in variables, and computers process values. For example, the computer can add integers, subtract integers, and do all the other basic arithmetic operations. Computers can also do all the basic arithmetic operations with double values.

Computers can even do all the basic operations on char values because chars are stored as integer values.

Special Rules about Integers

When you operate on two int values, you always get an int value. What if you divide 10 / 3 ? The "real" value is 3.3333... but in C++, the answer is 3. The reason is that the result of operating on two int values is always an int value. C++ just throws away the fractional part.

On the other hand, if one or both of the numbers has a fractional part, then the result will have a fractional part. Therefore:
has the value 3.333... because the bottom part has a decimal point.

**Special Rules about Strings**

Strings are different. If you write

"dog" + "cat"

the computer has no idea what the mathematical values you mean for the strings "dog" and "cat", so it does something different from adding numbers. In the case of strings the + operator tells the computer to stick the two strings together. In this case the result is "dogcat".

Can you subtract, multiply, or divide strings? No. It might make sense to say "cat" - "at", but C++ does not work that way.

And just to make things more complicated, you can add a char to a string and C++ will append the char to the string. Therefore

"rain" + 'y'

is the string "rainy"

**Summary**

Here are the basic types and their operations:

<table>
<thead>
<tr>
<th>Type</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>+ - * / %</td>
</tr>
<tr>
<td>double</td>
<td>+ - * /</td>
</tr>
<tr>
<td>char</td>
<td>+ - * / %</td>
</tr>
<tr>
<td>string</td>
<td>+</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type</th>
<th>&lt; &gt; &lt;= &gt;= == !=</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td></td>
</tr>
<tr>
<td>double</td>
<td></td>
</tr>
<tr>
<td>char</td>
<td></td>
</tr>
<tr>
<td>string</td>
<td></td>
</tr>
</tbody>
</table>