Date Types and Variables

A **data type** is a way to represent a particular set of values and to determine what operations can be performed on those values.

Just like there are different sets of numbers in mathematics (reals, non-reals, integers, whole, rational, irrational, etc) there are specific types of variables that can be used in a program.

There are 4 main types of data that are predefined in C++ but it is also possible to write special types that we will see later.

Integers: Simple numbers that do not contain fractional parts:

–10500
435
15
–25

The data types used to represent integers are *short, int,* and *long.*

Real Numbers: Numbers that include fractional parts:

–10.5
.435
15.
–25.1234

The data types used to represent real numbers are, *float, double,* and *long double.*

Booleans: The values *true* and *false*. These are used to make decisions, such as "if the first number is bigger than the second number, then switch the values."

The data types used to represent booleans is *bool.*

Characters: All of the symbols that can be produced by pressing the keys on the keyboard.

The data type *char* is used to represent characters. Each *char* is closed in apostrophes (single quotes) such as:

' A ' ' z ' ' 2 ' ' 9 ' ' * ' ' : '

Keep in mind that storing the character ' 1 ' is different than storing the integer 1. The character can't be used for mathematical calculations.

In addition to these characters we will soon talk about strings which are a collection of characters (words and sentences).

There are special symbols called **escape sequences**.

Escape Sequence	**Meaning**
‘\n’	linefeed
‘\b’	backspace
‘\t’	tab
‘\”\	double quote
‘\”	single quote
‘\\’	backslash

string Class

In addition to the built-in data types, C++ has the ability to use types that are defined in libraries. This is a main advantage to object oriented programming languages.

Two classes that we will use a lot are the *string* and *iostream* classes. There are more on each of these to come.

A *string* is a sequence of characters enclosed in quotation marks such as:

"A" "1234" "true" "Enter the distance in miles: "

Note that the string "A" is stored different than the character 'A' and "1234" can't be used in mathematical computation.

In later sections we will look at working more with strings to read, store, combine, and take apart strings. Any program that uses these advanced functions needs to contain the compiler directive

```
#include <string>
```

Purpose of Data Types

Using different data types help allow the compiler know what operations are valid on certain variables.

Also, since they are stored differently memory can be saved by choosing one data type over another.

Using an *int* type will generally use less memory than a *float* type.

Declarations

The **variable declarations** in a C++ program communite to the compiler the names of all variables used in the program.

They also tell the complier what kind of information will be stored in each variable.

A variable declaration begins with an identifier (for example, *float*) that tells the data type.

The declaration statement:

```
float miles,        // input: distance in miles
      kms;          // output: distance in kilometers
```

gives the names of two variables (miles, kms) used to store real numbers.

As usual the comments are ignored.

The statement

```
string lastName;
```

declares an identifier lastName fro storing a string.

Since *string* is a class, it is more accurate to refer to lastName as an object instead of a variable.

Constant Declarations

We often want to declare identifiers to be constants that do not change value.

This can be done by making a **constant declaration**.

The constant declaration:

```
const float KM_PER_MILE = 1.609;
```

sets KM_PER_MILE to have a constant value of 1.609. You will not be able to change this later in the program.

It makes it handy thought to change that value throughout the program by just changing the original declaration.