Local Variables

A **local variable** is one that occurs within a specific scope. They exist only in the function where they are created.

They are sometimes called ***automatic variables*** because they are automatically created when the function starts execution, and automatically go away when the function is finished executing.

The keyword **auto** can be used to explicitly create these variables, but isn't necessary since auto is the default.

Global Variables and extern

A **global variable** is a variable that is defined outside all functions and available to all functions.

These variables are unaffected by scopes and are always available, which means that a global variable exists until the program ends.

It is possible to create a global variable in one file and access it from another file. In order to do this, the variable must be declared in both files, but the keyword **extern** must precede the "second" declaration.

Static Variables

A **static variable** can be either a global or local variable. Both are created by preceding the variable declaration with the keyword **static**.

A **local static variable** is a variable that can maintain its value from one function call to another and it will exist until the program ends.

When a local static variable is created, it should be assigned an initial value. If it's not, the value will default to 0.

A **global static variable** is one that can only be accessed in the file where it is created. This variable is said to have **file scope**.

Constant Variables

In C, the preprocessor directive #define was used to create a variable with a constant value. This still works in C++, but problems could arise.

When #define is used, the preprocessor will go through the code and replace every instance of the #defined variable with the appropriate value. Well, since the #defined variable exists only in the file where it is created, it is possible to have the same definition in another file with a completely different value. This could lead to disastrous consequences.

To overcome this problem, the concept of a named constant that is just like a variable was introduced to C++.

To create a constant variable in C++, precede the variable declaration with the keyword **const**. This tells the compiler that "a variable has been created that has a value that cannot be changed"

When creating a constant variable, it MUST be assigned a value.