Creating a variable

In Java programming a "variable" is simply a useful container in which a value may be stored for subsequent use by the program. The stored value may be changed (vary) as the program executes its instructions – hence the term "variable".

A variable is created by writing a variable "declaration" in the program, specifying the type of data that variable may contain and a given name for that variable. For example, the **String** data type can be specified to allow a variable named "message" to contain regular text with this declaration:

```
String message ;
```

Variable names are chosen by the programmer but must adhere to certain naming conventions. The variable name may only begin with a letter, dollar sign $, or the underscore character _ , and may subsequently have only letters, digits, dollar signs, or underscore characters. Names are case-sensitive, so "var" and "Var" are distinctly different names, and spaces are not allowed in names.

Variable names should also avoid the Java keywords, listed in the table below, as these have special meaning in the Java language.

Each variable declaration must be terminated with a semi-colon character – like all other statements.

Strictly speaking, some words in this table are not actually keywords – **true**, **false**, and **null** are all literals, **String** is a special class name, **const** and **goto** are reserved words (currently unused). These are included in the table because they must also be avoided when naming variables.

abstract	**default**	**goto**	**package**	**synchronized**
assert	**do**	**if**	**private**	**this**
boolean	**double**	**implements**	**protected**	**throw**
break	**else**	**import**	**public**	**throws**
byte	**enum**	**instanceof**	**return**	**transient**
case	**extends**	**int**	**short**	**true**
catch	**false**	**interface**	**static**	**try**
char	**final**	**long**	**strictfp**	**void**
class	**finally**	**native**	**String**	**volatile**
const	**float**	**new**	**super**	**while**
continue	**for**	**null**	**switch**	

As good practice, variables should be named with words or easily recognizable abbreviations, describing that variable's purpose. For example, "button1" or "btn1" to describe button number one. Lowercase letters are preferred for single-word names, such as "gear", and names that consist of multiple words should capitalize the first letter of each subsequent word, such as "gearRatio" – the so-called "camelCase" naming convention.

Once a variable has been declared, it may be assigned an initial value of the appropriate data type using the equals sign = , either in the declaration or later on in the program, then its value can be referenced at any time using the variable's name.

Follow these steps to create a program that declares a variable, which gets initialized in its declaration then changed later:

1. Start a new program named "FirstVariable", containing the standard main method

```
class FirstVariable
{
        public static void main ( String[] args ) {        }
}
```

FirstVariable.java

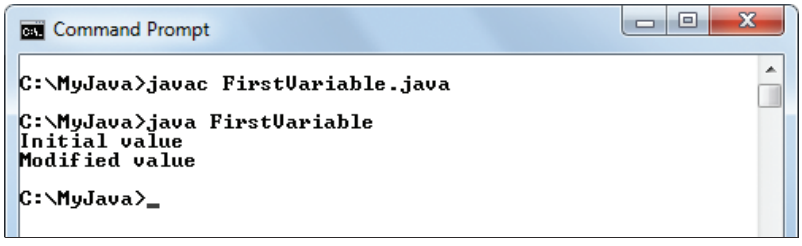
2. Between the curly brackets of the main method, insert this code to create, intialize, and output a variable

```
String message = "Initial value" ;
System.out.println( message ) ;
```

3. Add these lines to modify and output the variable value

```
message = "Modified value" ;
System.out.println( message ) ;
```

4. Save the program as **FirstVariable.java** then compile and run the program



If you encounter problems compiling or running the program you can get help from Troubleshooting Problems on page 22.