Recognizing data types

The most frequently used data types in Java variable declarations are listed in this table along with a brief description:

<table>
<thead>
<tr>
<th>Data type</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>char</td>
<td>A single Unicode character</td>
<td>‘a’</td>
</tr>
<tr>
<td>String</td>
<td>Any number of Unicode characters</td>
<td>“my String”</td>
</tr>
<tr>
<td>int</td>
<td>An integer number, from -2.14 billion to +2.14 billion</td>
<td>1000</td>
</tr>
<tr>
<td>float</td>
<td>A floating-point number, with a decimal point</td>
<td>3.14159265f</td>
</tr>
<tr>
<td>boolean</td>
<td>A logical value of either true or false</td>
<td>true</td>
</tr>
</tbody>
</table>

Notice that char data values must always be surrounded by single quotes and String data values must always be surrounded by double quotes. Also remember that float data values must always have an “f” suffix to ensure they are treated as a float value.

In addition to the more common data types above, Java provides these specialized data types for use in exacting circumstances:

<table>
<thead>
<tr>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>byte</td>
<td>Integer number from -128 to +127</td>
</tr>
<tr>
<td>short</td>
<td>Integer number from -32,768 to +32,767</td>
</tr>
<tr>
<td>long</td>
<td>Positive or negative integer exceeding 2.14 billion</td>
</tr>
<tr>
<td>double</td>
<td>Extremely long floating-point number</td>
</tr>
</tbody>
</table>

Specialized data types are useful in advanced Java programs – the examples in this book mostly use the common data types described in the top table.
Follow these steps to create a Java program that creates, initializes, and outputs variables of all five common data types:

1. Start a new program named “DataTypes” containing the standard main method
   ```java
   class DataTypes {
   public static void main ( String[] args ) {  }
   }
   ```

2. Between the curly brackets of the main method, insert these declarations to create and initialize five variables
   ```java
   char letter = 'M' ;
   String title = “Java in easy steps” ;
   int number = 365 ;
   float decimal = 98.6f ;
   boolean result = true ;
   ```

3. Add these lines to output an appropriate text String concatenated to the value of each variable
   ```java
   System.out.println( “Initial is “ + letter ) ;
   System.out.println( “Book is “ + title ) ;
   System.out.println( “Days are “ + number ) ;
   System.out.println( “Temperature is “ + decimal ) ;
   System.out.println( “Answer is “ + result ) ;
   ```

4. Save the program as DataTypes.java then compile and run the program

   ![Command Prompt](image)

   ```
   C:\MyJava>javac DataTypes.java
   C:\MyJava>java DataTypes
   Initial is M
   Book is Java in easy steps
   Days are 365
   Temperature is 98.6
   Answer is true
   C:\MyJava>_
   ```

The Java compiler will report an error if the program attempts to assign a value of the wrong data type to a variable – try changing the values in this example, then attempt to recompile the program to see the effect.