Type	Description	Wrapper Class
`byte`	8-bit signed 2s complement integer	`Byte`
`short`	16-bit signed 2s complement integer	`Short`
`int`	32-bit signed 2s complement integer	`Integer`
`long`	64-bit signed 2s complement integer	`Long`
`float`	32-bit IEEE 754 floating point number	`Float`
`double`	64-bit floating point number	`Double`
`boolean`	may be set to `true` or `false`	`Boolean`
`char`	16-bit Unicode (UTF-16) character	`Character`

Table 26.2.: Primitive types in Java

26.3.1. Declaration & Assignment

Java is a statically typed language meaning that all variables must be declared before you can use them or refer to them. In addition, when declaring a variable, you must specify both its type and its identifier. For example:

```
int numUnits;
double costPerUnit;
char firstInitial;
boolean isStudent;
```

Each declaration specifies the variable's type followed by the identifier and ending with a semicolon. The identifier rules are fairly standard: a name can consist of lowercase and uppercase alphabetic characters, numbers, and underscores but may *not* begin with a numeric character. We adopt the modern camelCasing naming convention for variables in our code. In general, variables *must* be assigned a value before you can use them in an expression. You do not have to immediately assign a value when you declare them (though it is good practice), but some value must be assigned before they can be used or the compiler will issue an error.[2]

The assignment operator is a single equal sign, `=` and is a right-to-left assignment. That is, the variable that we wish to assign the value to appears on the left-hand-side while the value (literal, variable or expression) is on the right-hand-size. Using our variables from before, we can assign them values:

[2] Instance variables, that is variables declared as part of an object *do* have default values. For objects, the default is `null`, for all numeric types, zero is the default value. For the `boolean` type, `false` is the default, and the default `char` value is `\0`, the null-terminating character (zero in the ASCII table).