Variables

and

Assignment Operators

Variables

A variable in JAVA and most computer programming languages is very different from the variables you are familiar with from Algebra. It is important to first make a clear delineation in your mind from the variables you use when working in math and those you use when writing computer code.

What is a computer programming variable then?

In short, a computer-programming variable (hereafter simply a "variable") is like a box that has been given a name. Once created, the box can be used to store information. There are essentially only four things that can be done to a variable:

1. Create the variable: In JAVA, this is done by specifying the variable's type and then giving it a name. This process is referred to as "Declaring a Variable."
2. Store a value in the variable using it's name to refer to it.
3. Retrieve the value currently stored in the variable, again by using it's name.
4. Forget about the variable. (More on this later!)

Let's consider the first item in the list. The first step when creating a variable is deciding what kind of information will be stored in it. In the JAVA programming language there are several primitive variable types, each one is designed to hold a specific type of information. For most programming purposes and for the AP Exam, the only primitive data types we are concerned with are:

- `int` – This data type holds integer values ranging from -2,147,483,648 to 2,147,483,647.
- `double` – This data type holds decimal values. The most important thing to remember about this variable type is that although it has an almost unlimited range of value, its accuracy is not guaranteed. Think of this like scientific numbers on your calculator, as the numbers get very large or very small accuracy is lost because only a certain number of digits are kept.
- `String` – This data type stores alphanumeric data. It is important to remember that a number that is stored as a String variable is still just a collection of characters, it will not be recognized as a number. Also notice that this variable type IS capitalized!

For a complete list of primitive JAVA variable types, visit the Oracle website:
http://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html

After deciding on the type our variable will be, next we need to pick a name for it. In JAVA, as with most programming languages, there are rules about how a variable can be named. The rules are fairly simple though:

1. It must start with a letter, the underscore character (_), or dollar sign character ($).
2. After the first character, the variable name can contain letters, underscores, dollar signs, and also numbers. (Note: Spaces are not permitted!)
3. The complete variable name must not be a JAVA keyword or reserved word. A complete list of such words can be found at:
   http://docs.oracle.com/javase/tutorial/java/nutsandbolts/_keywords.html

That covers the rules, but there are also guidelines commonly referred to as "Best Practices" that should also be followed.

Best Practices:

1. Even though the dollar sign and underscore are permitted, their use as the first character in a variable name is highly discouraged.
2. For now, the first character in your variable names should always be lower case.
3. Variables names should be descriptive if possible, i.e. `gear` is better than simply `g`, and `ratio` is better than just `r`.
4. When you want to use multiple words in the name, use an uppercase letter where you would normally put a space. For example, a variable called “Gear Ratio” would be written `gearRatio`. This practice is known as camel casing.

Now that we have established what the types are, the rules, and the best practices, it is time to examine a few examples:

Description of the variable	Variable Declaration in JAVA
A variable to hold the age of a student	`int studentAge;`
A variable to hold the name of a student	`String studentName;`
A variable to hold a student’s grade	`String studentGrade;`
A variable to record the date of enrollment at the school	(This one requires three variables)
	`int studentEnrollmentDay;`
	`int studentEnrollmentMonth;`
	`int studentEnrollmentYear;`
A variable to hold the student’s GPA	`double studentGPA;`

After a variable has been declared, a value can be stored in it using an “Assignment Operator.” In JAVA, the primary assignment operator is the equals character(=). Using the first example above (`studentAge`), a value can be given to the variable using its name and a value: `studentAge=16;`
To complete the example, let’s look at what the complete code would look like to fill in the following student information:

Name: Aaron
Age: 16
Grade: 10
Enrollment Date: August 6, 2013
GPA: 3.3

Code Example 1.1:

```
int studentAge;
String studentName;
int studentGrade;
int studentEnrollmentDay;
int studentEnrollmentMonth;
int studentEnrollmentYear;
double studentGPA;
studentAge=16;
studentName="Aaron";
studentGrade=10;
studentEnrollmentDay=6;
studentEnrollmentMonth=8;
studentEnrollmentYear=2013;
studentGPA=3.3;
```

It is also possible, and common, to combined variable declaration with an initial value assignment. The same code from *Example 1.1* written using combined declaration and assignment can be seen in *Example 1.2*.

Code Example 1.2:

```
int studentAge=16;
String studentName="Aaron";
int studentGrade=10;
int studentEnrollmentDay=6;
int studentEnrollmentMonth=8;
int studentEnrollmentYear=2013;
double studentGPA=3.3;
```

It is important to note that the equals assignment operator means "give the value after the equals to the variable that precedes it."

It is also possible to change the value of a variable after it has been declared by assigning it a new value (see *Example 1.3*).

In *Example 1.3* the variable `studentAge` is first given the value 16, then changed to 18 in the next line of code.

Code Example 1.3:

```
int studentAge=16;
studentAge=18;
```