

## **Java: Variables and Data Types**

Variables and data types are two important concepts in the Java programming language.

“Variables” are terms that represent a given piece of data, whether from the user or hard coded in the program. A “data type” refers to the type of data that is needed to record a given value. This handout will expound on what each of them is, how they work, and when to use them.

### **Data Types**

Data comes in all varieties and complexities. A data type could be as simple as a Boolean, which can only be one of two values: true or false. On the other hand, it could also be as complex as a video game character that involves hundreds of distinct pieces of data. This difference in complexity of values is why different data types exist and why it is important to understand the different data types. For instance, because computers handle whole numbers and decimals in different ways, 7 would be associated with an int while 7.0 would need to be a double.

Declaring the data type of a variable allows the computer to know how to work with the data. Because of this, understanding the fundamental data types is crucial to learning the Java programming language.

While some programming languages determine data types automatically based on the data provided, Java requires the programmer to explicitly declare what type of data a given variable will hold.

## Different Data Types and Their Uses

The following table lists several of the most common data types and examples of how they are used. Below the table are descriptions of the data types and examples of how they are used.

| Data Type | Example                                                  |
|-----------|----------------------------------------------------------|
| String    | <code>String x = "Hello World";</code>                   |
| char      | <code>char x = 'A';</code>                               |
| int       | <code>int x = 20;</code>                                 |
| double    | <code>double x = 20.5;</code>                            |
| boolean   | <code>boolean x = true;</code>                           |
| array     | <code>String[] x = {"apple", "Banana", "Cherry"};</code> |

- **String** is a sequence of characters. A String value is always enclosed in quotation marks (i.e., "apple"). In basic programs, Strings are used for input and output.
- **char** is used for single characters, and it is always enclosed in single quotation marks (i.e., 'a').
- **int**, short for integer, is used when the programmer wants the data to be represented by a whole numerical number.
- **double** is used to store a decimal value. Note: A decimal may be referred to as a "floating point number" in computer science.
- **boolean** holds a truth value, either `true` or `false`.
- **array** is a data type that can hold multiple pieces of data in a given order. An array must only contain members of the same data type.

## Variable Naming Rules

There are some restrictions when it comes to naming variables:

- Names can contain letters, numbers, underscores, and dollar signs.
- Names cannot begin with a number.
- Names cannot contain whitespace.
- Reserved words (like Java keywords, such as `int` or `Boolean`) cannot be used as names.

*Note: Remember that all Java code, including variable names, is case sensitive.*

Provided by

Java: Variables and Data Types

## Variable Naming Guidelines

Some important guidelines to ensure variables are readable and understandable to fellow programmers (or your future self) include:

- Names should be descriptive and relevant to the variable's purpose.
- Names should follow consistent conventions, typically camel case in Java (i.e., "userResponse", "speedOfLight", etc.).

## Using Variables

After a variable has been declared and initialized, it can be used to compute the values of other variables.

### Example:

```
9 // variableName = value;  
10 int num1 = 3;  
11 int num2 = 12;  
12 int product = num1 * num2;
```

In the example code, there are three variables. Two of them are assigned values directly, and the third is assigned the value of the first two multiplied together. If the programmer decided to print the third variable, the number 36 would be displayed.

## Declaration and Assignment

Creating a variable and assigning it a value requires 4 parts:

1. **Data type:** This is the type of data that the variable will hold.
2. **Variable name:** The variable name will act as an alias for the assigned value, allowing the programmer to reference the name when they need to work with the value.
3. **Assignment operator:** The equal sign signals that new data is being stored in the variable.
4. **Value:** This is the data that will be stored in the variable.

Creating a variable is called **declaring** that variable. The variable below is declared but does not yet hold a value.

**Data type** `int` **Variable Name** `number;`

Associating a variable with data is called **assignment**. Similarly, **initialization** occurs the first time a value is assigned to a new variable. Below, the existing variable is initialized with the value 20.

**Variable Name** `number` **Assignment Operator** `=` **Value** `20;`

It is possible to declare and initialize a variable in a single line as shown below.

**Data type** `int` **Variable Name** `number` **Assignment Operator** `=` **Value** `20;`

The Academic Center for Excellence (ACE) offers free on-campus or online tutoring appointments. For further assistance with programming concepts, please call ACE at (540) 891-3017 or email us at [ACE@germanna.edu](mailto:ACE@germanna.edu).