Task: Writing First Java Program

www.hyperiondev.com

Introduction

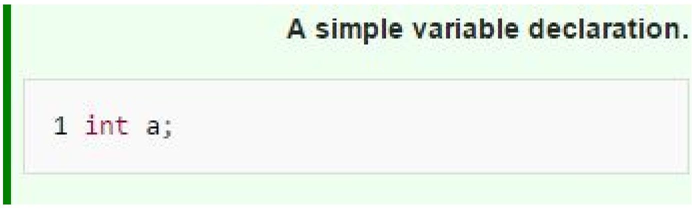
Welcome to the Writing First Java Program Task
Overview:

Now that you know how to compile and run Java programs it's time to write one! You'll gain knowledge on variable types and their declarations as well as variable manipulation. An introduction to data types will be covered, giving you some exposure to working with strings to create a program output which is is beneficial in some way and can be understood by the user.

- The Hyperion Team

A note from the Hyperion Team...

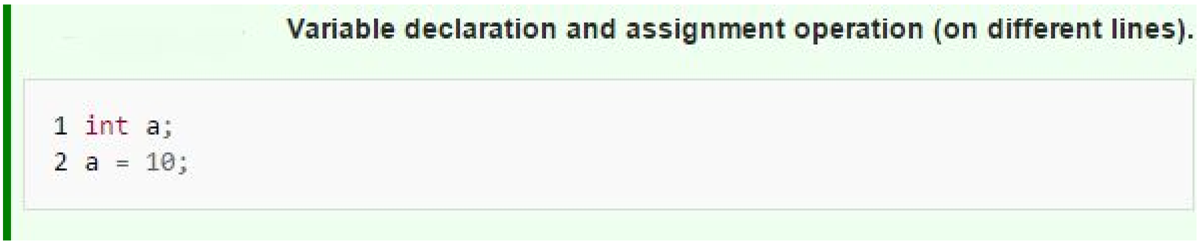
The following code demonstrates how a simple variable can be created. This process is known as *variable declaration*.



A simple variable declaration.

```java
int a;
```

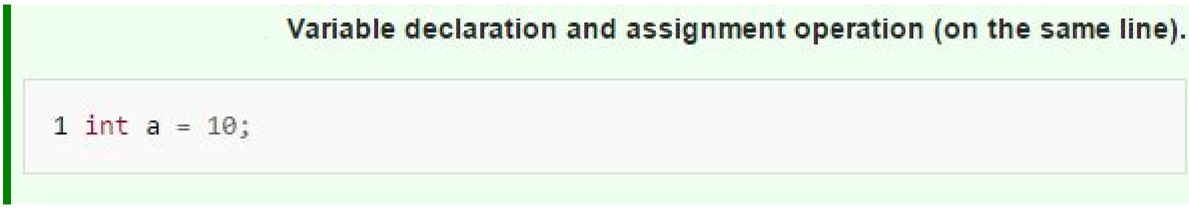
Once a variable is created, we can provide it with any integer value using the following syntax. This process is called an *assignment operation*.



Variable declaration and assignment operation (on different lines).

```java
int a;
a = 10;
```

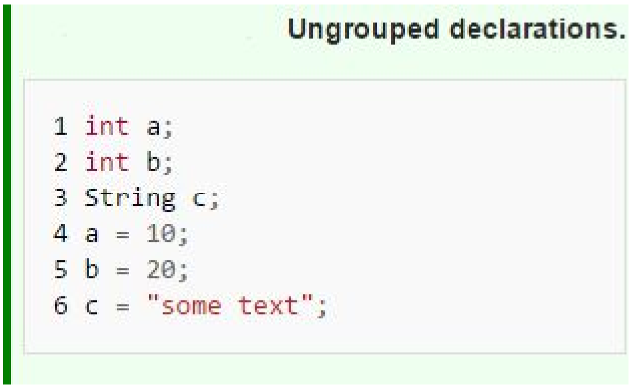
Java provides programmers with a simpler way of combining both variable declaration and assignment operation in one line. Consider the following code:



Variable declaration and assignment operation (on the same line).

```java
int a = 10;
```

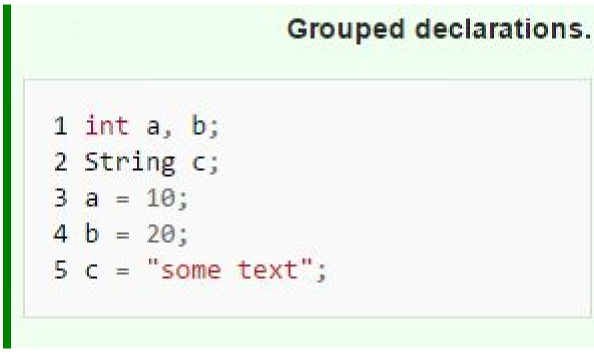
Consider the following code:



Ungrouped declarations.

```java
int a;
int b;
String c;
a = 10;
b = 20;
c = "some text";
```

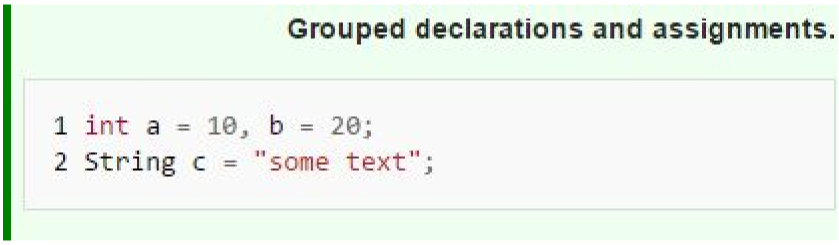
You can group the declarations of similar data types in one statement, for instance:



Grouped declarations.

```java
int a, b;
String c;
a = 10;
b = 20;
c = "some text";
```

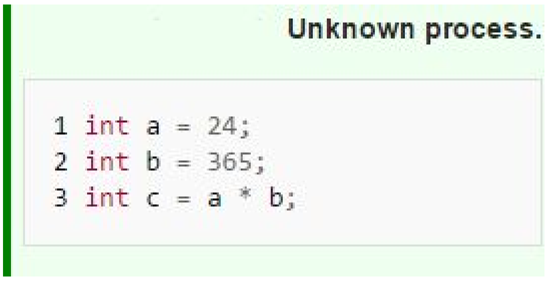
You can further reduce the syntax by doing group declarations and assignments together, as such:



Grouped declarations and assignments.

```java
int a = 10, b = 20;
String c = "some text";
```

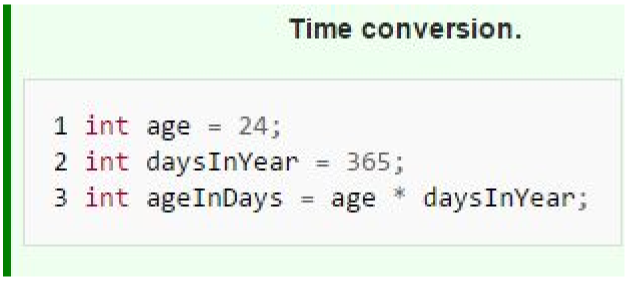
When naming identifiers, we should always name our variables in a way that tells us what they hold. Consider this example:



Unknown process.

```
int a = 24;
int b = 365;
int c = a * b;
```

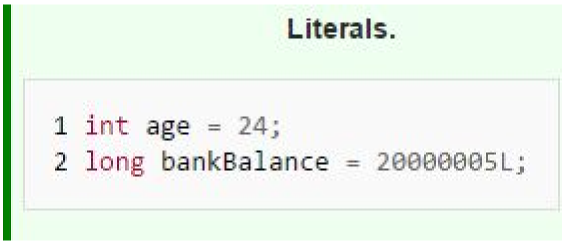
Do you know what this program does? Well, it multiplies two values. That much you guessed right. But, do you know what those values are? Exactly, you don't. Now consider this code:



Time conversion.

```
int age = 24;
int daysInYear = 365;
int ageInDays = age * daysInYear;
```

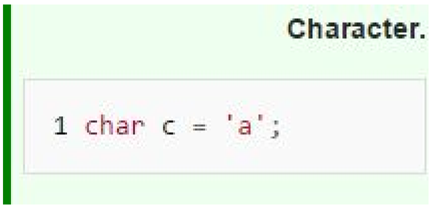
Simple values like numbers are called *literals*. Consider the following code:



Literals.

```
int age = 24;
long bankBalance = 20000005L;
```

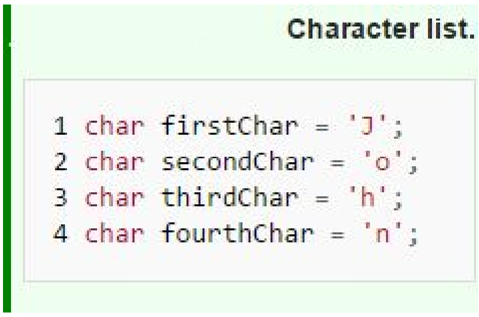
Characters are basically letters of the English alphabet. When writing a single character, we use single quotes to encapsulate them. Take a look at the code below:



Character.

```
char c = 'a';
```

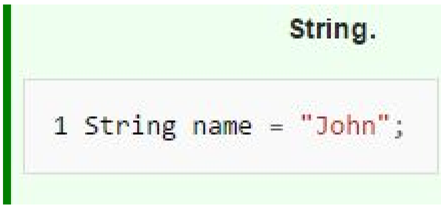
What if you had to store a complete name within them, say *John*, would you write something like:



Character list.

```
char firstChar = 'J';
char secondChar = 'o';
char thirdChar = 'h';
char fourthChar = 'n';
```

No ways! Thankfully, there's a data type that handles large number of characters, it's called a *string*. A string can be initialized as follows:



String.

```
String name = "John";
```

Instructions

First read example.java, open it using jGRASP (Right click the file and select 'Open with jGRASP").

- In this folder there is a file called example.java
- Open this folder using the JGRASP program. You can either do this by right clicking on the example.java file, going to the 'Open with' menu, and selecting JGrasp.exe. Alternatively, you can run the JGRASP program on your computer, go to the top left corner of the program, select File->Open and navigate to example.java on your hard drive and double click on it to open it.
- Once example.java is open in JGRASP please read all of its content very carefully. This will help you understand the basic structure of a Java program.
- There is a compulsory task at the end of the example.java document. The instructions of what to do to complete the task can be found here. You must complete this exercise to continue onto the next task of this course.

Compulsory Task

Follow these steps:

- Open task2.java in your task folder.
- Using the examples show in the example.java file, write code in task2.java to do the following:
    - Create four variables, using the appropriate data type (string, double, int, or char) and have them store:
        - Your first name
        - Your age
        - Your approximate height in centimetres (decimal)
        - Your favourite whole number
- Output this data on separate lines with tabbed spacing (eg: First name: Joe)
- You'll need to use four separate system.out.println statements to do this, each adding your variable to a string to output them.
- Don't forget to compile and run your code to test the output

Optional Task

Follow these steps:

- Open task2.java in your task folder
- You should make the following modifications to your original program:
    - Change your first name to your full name, with correct formatting.
    - Add 5 years to your age using the add operator.
    - Divide your favourite whole number by 3 - briefly explain in a comment what the expected result was, why it was wrong, and how this could be remedied.

Things to look out for

1. Make sure that you have installed and setup all programs correctly. You have setup **Dropbox** correctly if you are reading this, but **jGRASP or Java** may not be installed correctly.
2. If you are not using Windows, please ask your tutor for alternative instructions.

Still need help?

Just write your queries in your comments.txt file and your tutor will respond.

Task Statistics

Last update to task: 19/01/2016.
Authors: Riaz Moola and Jared Ping.
Main trainer: Umar Randeree.
Task Feedback link: Hyperion Development Feedback.