**Algorithms and Java basics:**

variables, assignment, interactive programs, pseudocode

CSC 1051 – Algorithms and Data Structures I

Dr. Mary-Angela Papalaskari
Department of Computing Sciences
Villanova University

Course website:
www.csc.villanova.edu/~map/1051/

---

**Some types of data in Java**

<table>
<thead>
<tr>
<th>type</th>
<th>set of values</th>
<th>literal values</th>
<th>operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>char</td>
<td>characters</td>
<td>&quot;A&quot;</td>
<td>compare</td>
</tr>
<tr>
<td>String</td>
<td>sequences of characters</td>
<td>&quot;Hello World&quot;</td>
<td>concatenate</td>
</tr>
<tr>
<td>int</td>
<td>integers</td>
<td>17</td>
<td>add, subtract, multiply, divide, mod</td>
</tr>
<tr>
<td>double</td>
<td>floating-point numbers</td>
<td>3.1415</td>
<td>add, subtract, multiply, divide</td>
</tr>
<tr>
<td>boolean</td>
<td>truth values</td>
<td>true</td>
<td>and, or, not</td>
</tr>
</tbody>
</table>

---

**Assignment Statement**

- **Changes the value of a variable**
- The assignment operator is the `=` sign

- The expression on the right is evaluated and the result is stored in the variable on the left
Assignment Statement

A variable can be given an initial value in the declaration
- a new value can be assigned later:

```java
int age = 18;
double x = 3.2, y = -0.80;
String name = scan.nextLine();
age = 19;
x = x + 0.5;
name = scan.nextLine();
```

Assignment Statement

A variable can be given an initial value in the declaration
- a new value can be assigned later:

```java
int age = 18;
double x = 3.2, y = -0.80;
String name = scan.nextLine();
int age = 19;
```

Variables are only declared ONCE

Tracing the values of variables after each statement.

```
int age = 18;
double x;
String name = "Sherlock";
age = 19;
x = 0.5;
x = x + 0.2;
name = name + "Holmes";
```

Trace: TRY THIS:

```
int a, b;
a = 3;
b = 4;
int c = a;
a = b;
b = 5;
b = c;
```

Final values:

```
a b c
```

Dr Papalaskari
CONSTANTS: like variables, but value cannot change – declare using `final` modifier:

```java
final int INCHES_PER FOOT = 12;
final double LBS_PER KG = 2.2;
```

Convention: Use UPPER_CASE identifiers

Arithmetic Operators

- Addition +
- Subtraction -
- Multiplication *
- Division /
- Remainder %

- If either or both operands used by an arithmetic operator are floating point (e.g., type `double`), then the result is a floating point

Example

```java
int feet = 25;
int inches = feet * INCHES_PER FOOT;

int seconds = 143;
int minutes = seconds / 60;
int remainingSeconds = seconds % 60;
```

Division and Remainder

- If both operands are integers (e.g., type `int`), the division result is an integer (the fractional part is discarded):

<table>
<thead>
<tr>
<th>Integer Division</th>
<th>Remainder</th>
</tr>
</thead>
<tbody>
<tr>
<td>14 / 3</td>
<td>14 % 3</td>
</tr>
<tr>
<td>143 / 60</td>
<td>143 % 60</td>
</tr>
<tr>
<td>20 / 16</td>
<td>20 % 16</td>
</tr>
<tr>
<td>8 / 12</td>
<td>8 % 12</td>
</tr>
</tbody>
</table>
Operator Precedence

**What is the order of evaluation of sub-expressions?**

1. Multiplication, division, remainder
2. addition, subtraction, string concatenation

- Operators with the same precedence: left → right
- Use parentheses to override default order

**more examples:**

```
result = total + count / max - offset;
```

```
a + b + c + d + e
```

```
a / (b + c) - d % e
```

```
a - b / c + d * e
```

```
a / (b * (c + (d - e)))
```

Increment and Decrement

- The *increment operator* (++) adds one to its operand
- The *decrement operator* (--) subtracts one from its operand
- The statement `count++;` is functionally equivalent to `count = count + 1;`

Algorithms in everyday life

**Algorithms**

An algorithm is a specific set of instructions for carrying out a procedure or solving a problem, usually with the requirement that the procedure terminate at some point. Specific algorithms sometimes also go by the name *method*, *procedure*, or *technique*. The word "algorithm" is a distortion of *al-Khwārizmī* [named after Muhammad ibn al-Khwārizmī], a Persian mathematician who wrote an influential treatise about algebraic methods.

Sources: [http://mathworld.wolfram.com/Algorithm.html](http://mathworld.wolfram.com/Algorithm.html) and [Wikipedia](http://en.wikipedia.org/wiki/Muhammad_ibn_M%C5%ABs%C4%81_al-Khw%C4%81rizm%C4%AB)
Algorithm Example: 
**Input-Compute-Output** pattern

**GPA problem:** Write a program that computes and outputs the GPA, given the credits and quality points earned.

**Variables:** qp, credits, gpa

**Algorithm:**
1. qp = input from user
2. credits = input from user
3. gpa = qp / credits
4. Print gpa

Java Program →

**Algorithm**

**Variables:** qp, credits, gpa

**Algorithm:**
1. qp = input from user
2. credits = input from user
3. gpa = qp / credits
4. Print gpa

Interactive Programs – Input/Output

• Programs can use data obtained during runtime, eg:

```java
int age;
String name;

System.out.print("Enter your name");
name = scan.next();
System.out.print("Enter your age");
age = scan.nextInt();
```

Interactive Programs – Input/Output

• In Java, you first need to create a Scanner object

```java
int age;
String name;
Scanner scan = new Scanner(System.in);
System.out.print("Enter your name");
name = scan.next();
System.out.print("Enter your age");
age = scan.nextInt();
```
Interactive Programs – Input/Output

- The Scanner class is part of the java.util class library, and must be imported into a program in order to be used.
- The import statement goes at beginning of your program (above class definition).

```java
import java.util.Scanner;

public class GPA {
    public static void main (String[] args)
    { //-------------
        //-------------
    }
}
```

Interactive Programs – Input/Output

**Summary:**
1. import the Scanner class, i.e., add this before the class definition of your program:
   ```java
   import java.util.Scanner;
   ```
2. In your main method, before doing any input, declare and initialize the Scanner object:
   ```java
   Scanner scan = new Scanner(System.in);
   System.out.print("Enter your name");
   name = scan.next();
   System.out.print("Enter your age");
   age = scan.nextInt();
   ```
3. Input away!

**Scanner methods**
- `nextInt()` → input an int
- `nextDouble()` → input a double
- `nextLine()` → input a String (until end of line)
- `next()` → input a String `token` (one word or other delimited “chunk” of text)
  - White space (space, tab, new line) are used to separate input tokens

```java
import java.util.Scanner;
public class TellMeAboutYou
{
    public static void main(String[] args)
    {
        int age;
        String name;
        Scanner scan = new Scanner(System.in);
        System.out.print("Enter your name");
        name = scan.next();
        System.out.print("Enter your age");
        age = scan.nextInt();
        System.out.println("Pleased to meet you, " + name + "!");
    }
}
```

Example

Enter your name: Fiona
Enter your age: 17
Pleased to meet you, Fiona!

Inspired by: http://www.onlineconversion.com/dogyears.htm
Using the `Scanner` class

```java
import java.util.Scanner;

public class TellMeAboutYou {
    public static void main(String[] args) {
        int age;
        String name;
        Scanner scan = new Scanner(System.in);
        // input name AND age
        name = scan.nextLine();
        // Display appropriate message
        System.out.println("Pleased to meet you, " + name + ", Fiona!");
    }
}
```

More examples – see text: `Echo.java`, `GasMileage.java`

---

**Writing an algorithm in pseudocode**

- List the variables used.
- List the steps for solving the problem, in order.
- Try to be brief and unambiguous; use Java expressions only when it is simpler to specify a step in Java than in English.

**Variables:** `qp`, `credits`, `gpa`

**Algorithm:**
1. `qp` = input from user
2. `credits` = input from user
3. `gpa` = `qp` / `credits`
4. Print `gpa`

---

**Example**

Write an algorithm to solve the following problem:

*Input values representing a time duration in hours, minutes, and seconds and then calculate and output the equivalent total number of seconds.*
Example
Write an algorithm to solve the following problem:

*Input a value representing a number of seconds, then calculate and output the equivalent amount of time as a combination of hours, minutes, and seconds.*