

## Second Java Program

```
import java.util.Scanner;

public class EggBasket
{
    public static void main(String[] args)
    {
        int numberOfBaskets, eggsPerBasket, totalEggs;

        Scanner keyboard = new Scanner(System.in);

        System.out.print(
            "Enter the number of eggs in each basket: ");
        eggsPerBasket = keyboard.nextInt();
        System.out.print("Enter the number of baskets: ");
        numberOfBaskets = keyboard.nextInt();

        totalEggs = numberOfBaskets * eggsPerBasket;

        System.out.println(eggsPerBasket + " eggs per basket.");
        System.out.println(numberOfBaskets + " baskets.");
        System.out.println("Total number of eggs is " + totalEggs);
    }
}
```

## What Does EggBasket Do?

- Take a look at the program and see if you can figure out what it does.

## What Is a Program Variable?

```
int numberOfBaskets, eggsPerBasket, totalEggs;
```

- This is a *declaration* of three integer variables
- A **variable** is a named location to store data, i.e., a container for data
- Each variable can hold only one type of data; for example only integers, only floating point (real) numbers, or only characters
- All program variables **must** be *declared* before being used

## What Is a Program Type?

- A variable's **type** determines the kind of values that a variable can hold and what operations can be applied to it.
- Some Java *primitive* types:
  - **int** (integer, whole values, e.g., 0, 1, -13, 231)
  - **double** (real values, e.g., 0.0, 3.1415, -2.72)
  - **char** (single character values, e.g., 'a', '3', '\$')
  - **boolean** (only one of two values: **true**, **false**)

## How Do We Assign/Change the Value of a Variable?

```
eggsPerBasket = keyboard.nextInt();  
totalEggs = numberOfBaskets * eggsPerBasket;
```

- **Assignment statement:**  
`variable = expression;`
- Assigns the value of the expression on the right side of = to the variable on the left side
- It does not mean “equal” like in math!

## What Is an Expression?

```
numberOfBaskets * eggsPerBasket
```

- Program expressions are very much like arithmetic expressions you are familiar with (usual operators, parenthesis, precedence rules, etc.)
- Expressions can be evaluated to produce a value and they have a type (the type of the value of the expression)

## Numeric Operators

- Some common integer operators:
  - + (obvious)
  - - (obvious)
  - \* (obvious)
  - / (integer division, e.g.,  $6/2=3$ ,  $5/2=2$ ,  $19/5=?$ )
  - % (mod operator, i.e., remainder of integer division, e.g.,  $6\%2=0$ ,  $5\%2=1$ ,  $19\%5=?$ )
- Some common real operators:
  - +, -, \*, / (real division)

## Some Expressions: What Are Their Values?

```
int i = 12, j = 5, k = -3;  
double x = 2.1, y = -1.5, z = 3.0;
```

- $(i + j + k) / 3$
- $(i / j) * j + (i \% j)$
- $x * x + y * y$
- $(x + y + z) / (x - y - z)$
- $2.0 * z - (x + y)$

# Output Statements

```
System.out.println(output1 + output2 + ... + outputN);  
System.out.print(output1 + output2 + ... + outputN);
```

- Concatenates the various outputs (quoted strings, variables, constants/numbers) and prints them to the screen (*println* adds a newline).
- What do the following statements output?

```
int day = 19;  
System.out.print("January " + day + ", " + 2004 +  
    " is Martin Luther King's Day! ");
```

# Input Statements

Given the declaration:

```
Scanner in = new Scanner(System.in);
```

- Declare and input an integer value:  

```
int i = in.nextInt();
```
- Declare and input a real value:  

```
double x = in.nextDouble();
```
- Declare and input a whole line (a string of characters):  

```
String s = in.nextLine();
```
- Input of a single character with the Scanner class is trickier; we'll see it later.

## To Sum Up...

- So far, we have seen
  - how to *input* values from the keyboard
  - how to *output* messages/values to the screen
  - how to create variables to store values
  - how to store values in variables and compute new values with expressions

## It's Your Turn!

- Now let's put it all together: Write a Java program called *ComputeArea*, which asks the user for the *width* and *height* of a rectangle and computes and prints the area of the rectangle.

# Complete ComputeArea

## The `char` Type

- A variable of the `char` data type stores a single “printable” character
- A `char` constant or literal is a character in single quotes, e.g., `'y'`
- For example:

```
char answer = 'y';  
System.out.println(answer);  
prints (displays) the letter y
```