Data and Variables

Mitsu Ogihara

Department of Computer Science
University of Miami
Table of Contents

1. Data, Literals, and Variables
2. Using Variables
3. Using Variables for Computation
4. Binary Operations on Numbers
The programs we have seen so far used only:

- Method calls and
- System.out.println and System.out.print

We will learn now how to record and modify information during execution of a program.
Data

- The bit is the fundamental unit in computation
- The bit has two values, 0 and 1 ("off" and "on")
- In computers and in computer programs, information is encoded as a finite sequence of bits, and that is called **data**
What Are Data Like to a Program?

- Data come and go
- Data require specific ways of interpretation (type)
  - The same sequence of bits may be interpreted differently according to the type
- Data can be generated by operations
- Data can be stored for future references $\rightarrow$ variable
- Data can be succinctly specified $\rightarrow$ literal
Variables and Literals

- A **variable** is a place to store data during execution of a program
- Because data must have a type and a value, so must a variable
- On the other hand, **literals** are data with values but no name

For example, the "Hello, World!" as it appears in the statement

```java
System.out.println("Hello, World!")
```

is a literal
A **primitive data type** in Java is a data type with a fixed number of bits allocated for storing information. There are four primitive data types for whole numbers in Java:

- **byte**: 8 bits; $-128$ through $127$
- **short**: 16 bits; $-32,768$ through $32,767$
- **int**: 32 bits; $-2,147,483,648$ through $2,147,483,647$
- **long**: 64 bits; $-9,223,372,036,854,775,808$ through $9,223,372,036,854,775,807$

There are two primitive data types for floating numbers (real numbers; i.e., specifications for digits below the decimal point):

- **float**: 32 bits; $3.4 \times 10^{-38}$ through $3.4 \times 10^38$
- **duble**: 64 bits; $1.7 \times 10^{-308}$ through $1.7 \times 10^{308}$

There are also **boolean** (one bit, logical value) and **char** (16 bits, a character).
Number Literals

In Java an exact number can be specified by providing the digits and by default such a number is thought of as either an int or a double; e.g.,

- 401 (as an int)
- \(-2.223344\) (as a double)

Such specification fails if the number goes out of the range of the data type, e.g., 9876543210 (more than 32 bits will be needed)

These are called **number literals**
Number Arithmetic

In a formula

\[ \frac{27}{10} \]

27 and 10 are **operands** and / is an **operator**

/ is an operator that takes two operands and so is a **binary operator**

There are five **binary operators**: +, −, *, /, and %

% is the remainder that preserves the sign of the number to be divided;

\[ -40.5 \% \ 2 = -0.5, \ \ 0 \% \ 3 = 2 \]

If both operands are integers / is the quotient; that is,

\[ 10 \ / \ 3 = 1 \]
Arithmetic Resolution

*, /, and % have precedence over + and −
If one of the operands is a double, the result will be a double

\[
27 - (10 + 4.5 \times 2) + (-9.0/2 \% 2) \\
= 27 - (10 + 9.0) + (-9.0/2 \% 2) \\
= 27 - 19.0 + (-9.0/2 \% 2) \\
= 8.0 + (-9.0/2 \% 2) \\
= 8.0 + (-4.5 \% 2) \\
= 8.0 + (-0.5) \\
= 7.5
\]
Viewing the Value via System.out.println

System.out.println and System.out.print print the value of a number literal or a number literal formula

```java
public class NumberFormulas {
    public static void main( String[] args ) {
        System.out.print( "-40.5 % 2 is " );
        System.out.println( -40.5 % 2 );
        System.out.print( "20 % 3 is " );
        System.out.println( 20 % 3 );
        System.out.print( "27 - (10 + 4.5 * 2) + (-9.0 / 2 % 2) is " );
        System.out.println( 27 - (10 + 4.5 * 2) + (-9.0 / 2 % 2) );
        System.out.print( "10 / 3 is " );
        System.out.println( 10 / 3 );
        System.out.print( "10 / 3.0 is " );
        System.out.println( 10 / 3.0 );
    }
}
```

Notice the output of print is finite and may not agree with with our perception at the last digit
Strings can be connected with the ’+’ sign, which means **concatenation**.
If either side of a ’+’ sign is a String the result is a String.

```java
public class StringConcat {
    public static void main( String[] args ) {
        System.out.print( ""abc" + "def" is " );
        System.out.println( "abc" + "def" );
        System.out.print( "0 + 1 is " );
        System.out.println( 0 + 1 );
        System.out.print( "0 + 1 + "234" is " );
        System.out.println( 0 + 1 + "234" );
        System.out.print( "0 + "1\" + 234 is " );
        System.out.println( 0 + "1\" + 234 );
        System.out.print( "0 + "1\" + "234\" is " );
        System.out.println( 0 + "1\" + "234\" );
    }
}
```
Basic Actions on a Variable

1. Declaring a variable with its type (declaration)
   `<type> <name>`

2. Assigning a value to a variable (assignment)
   `<name> = <value>`
   RHS can be a formula; the value is evaluate and then given to the variable

3. Obtaining the value held by a variable (reference)
   `<name>`

4. Passing it to a method
   `<method_name>(<name>)`

5. In the case of a non-primitive (that is, object) data type, perform one of its permissible method
   `<name>..<method_name>(<parameter>)`

- Declaration should precede reference or assignment
- Actions other than declaration can be performed any number of times
The String Type

```java
public class HelloWorldString {
    public static void main( String[] args ) {
        String message;
        message = "Hello, World!";
        System.out.println( message );
        message = "Hello, Class!";
        System.out.println( message );
        message = "Welcome to the world of Java!";
        System.out.println( message );
    }
}
```

Declaration of the String variable `message`
The String Type

```java
public class HelloWorldString {
    public static void main( String[] args ) {
        String message;
        message = "Hello, World!";
        System.out.println( message );
        message = "Hello, Class!";
        System.out.println( message );
        message = "Welcome to the world of Java!";
        System.out.println( message );
    }
}
```

Three different assignments to the variable
The String Type

```java
public class HelloWorldString {
    public static void main( String[] args ) {
        String message;
        message = "Hello, World!";
        System.out.println( message );
        message = "Hello, Class!";
        System.out.println( message );
        message = "Welcome to the world of Java!";
        System.out.println( message );
    }
}
```

Passing the variable to `System.out.println` to have its value printed on screen
A natural extension of printing a String literal
Using Multiple Strings

```java
public class HelloStrings {
    public static void main( String[] args ) {
        String helloMessage;
        String nameMessage;
        String loveMessage;

        helloMessage = "Hello, World!";
        nameMessage = "My name is Mitsu!";
        loveMessage = "I love computing.";

        System.out.println( helloMessage );
        System.out.println( nameMessage );
        System.out.println( loveMessage );
    }
}
```

Declaration of the String variables
Using Multiple Strings

```java
public class HelloStrings {
    public static void main( String[] args ) {
        String helloMessage;
        String nameMessage;
        String loveMessage;

        helloMessage = "Hello, World!";
        nameMessage = "My name is Mitsu!";
        loveMessage = "I love computing.";

        System.out.println( helloMessage );
        System.out.println( nameMessage );
        System.out.println( loveMessage );
        System.out.println( helloMessage );
        System.out.println( nameMessage );
        System.out.println( loveMessage );
    }
}
```

Assignment to the variables
Using Multiple Strings

```java
public class HelloStrings {
    public static void main( String[] args ) {
        String helloMessage;
        String nameMessage;
        String loveMessage;
        helloMessage = "Hello, World!";
        nameMessage = "My name is Mitsu!";
        loveMessage = "I love computing.";
        System.out.println( helloMessage );
        System.out.println( nameMessage );
        System.out.println( loveMessage );
        System.out.println( helloMessage );
        System.out.println( nameMessage );
        System.out.println( loveMessage );
    }
}
```

Print the messages
Using Multiple Strings

```java
public class HelloStrings {
    public static void main( String[] args ) {
        String helloMessage;
        String nameMessage;
        String loveMessage;
        helloMessage = "Hello, World!";
        nameMessage = "My name is Mitsu!";
        loveMessage = "I love computing.";
        System.out.println( helloMessage );
        System.out.println( nameMessage );
        System.out.println( loveMessage );
        System.out.println( helloMessage );
        System.out.println( nameMessage );
        System.out.println( loveMessage );
    }
}
```

Print the messages again
Scope of a Variable

For a variable declared within a method, the name is valid between

- the point of declaration and
- the close-curly-bracket ’}’ of the inner-most matching pair of curly brackets that include the declaration
Table of Contents

1. Data, Literals, and Variables
2. Using Variables
3. Using Variables for Computation
4. Binary Operations on Numbers
Compute Various Values Given a Radius

Given a radius $R$, compute the following:

- The perimeter of a circle having radius $R \ldots 2\pi R$
- The area of a circle having radius $R \ldots \pi R^2$
- The surface area of a sphere having radius $R \ldots 4\pi R^2$
- The volume of a sphere having radius $R \ldots \frac{4}{3}\pi R^3$
The Program

```java
// compute values given a radius
public class Radius {
    public static void main( String[] args ) {
        int radius;
        double circlePerimeter, circleArea, ballArea, ballVolume;
        double pi;

        //--- set the values of pi and radius
        pi = 3.14159265;
        radius = 10;
        // calculate the values
        circlePerimeter = 2 * pi * radius;
        circleArea = pi * radius * radius;
        ballArea = 4 * pi * radius * radius;
        ballVolume = 4 * pi * radius * radius * radius / 3;
    }
}
```

Declare `radius` to be an `int` variable and `circlePerimeter`,
`circleArea`, `ballArea`, `ballVolume`, `pi` to be `double` variables.

To declare multiple variables of the same type, you may use a comma
between the names:

 `<type> <name1>, <name2>, ... <namek>;`
The Program

```java
// compute values given a radius
public class Radius {
    public static void main( String[] args ) {
        int radius;
        double circlePerimeter, circleArea, ballArea, ballVolume;
        double pi;
        //--- set the values of pi and radius
        pi = 3.14159265;
        radius = 10;
        // calculate the values
        circlePerimeter = 2 * pi * radius;
        circleArea = pi * radius * radius;
        ballArea = 4 * pi * radius * radius;
        ballVolume = 4 * pi * radius * radius * radius / 3;
    }
}
```

pi and radius are assigned data from the literals 3.14159265 and 10
The Program

```java
// compute values given a radius
public class Radius {
    public static void main( String[] args ) {
        int radius;
        double circlePerimeter, circleArea, ballArea, ballVolume;
        double pi;
        //--- set the values of pi and radius
        pi = 3.14159265;
        radius = 10;
        // calculate the values
        circlePerimeter = 2 * pi * radius;
        circleArea = pi * radius * radius;
        ballArea = 4 * pi * radius * radius;
        ballVolume = 4 * pi * radius * radius * radius / 3;
    }
}
```

The values for `circlePerimeter`, `circleArea`, `ballArea`, `ballVolume` receive values from calculation
Here `*` is the multiplication
The Program (cont’d)

```java
//-- output the values
System.out.print( "The radius is " );
System.out.println( radius );
System.out.print( "The perimeter is " );
System.out.println( circlePerimeter );
System.out.print( "The area of the disc is " );
System.out.println( circleArea );
System.out.print( "The area of the ball is " );
System.out.println( ballArea );
System.out.print( "The volume of the ball is " );
System.out.println( ballVolume );
```

Print the radius
The Program (cont’d)

```java
//-- output the values
System.out.print( "The radius is " );
System.out.println( radius );
System.out.print( "The perimeter is " );
System.out.println( circlePerimeter );
System.out.print( "The area of the disc is " );
System.out.println( circleArea );
System.out.print( "The area of the ball is " );
System.out.println( ballArea );
System.out.print( "The volume of the ball is " );
System.out.println( ballVolume );
```

Print the perimeter
The Program (cont’d)

```java
//--- output the values
System.out.print( "The radius is " );
System.out.println( radius );
System.out.print( "The perimeter is " );
System.out.println( circlePerimeter );
System.out.print( "The area of the disc is " );
System.out.println( circleArea );
System.out.print( "The area of the ball is " );
System.out.println( ballArea );
System.out.print( "The volume of the ball is " );
System.out.println( ballVolume );
```

Print the area of the disk
The Program (cont’d)

```java
//-- output the values
System.out.print( "The radius is " );
System.out.println( radius );
System.out.print( "The perimeter is " );
System.out.println( circlePerimeter );
System.out.print( "The area of the disc is " );
System.out.println( circleArea );
System.out.print( "The area of the ball is " );
System.out.println( ballArea );
System.out.print( "The volume of the ball is " );
System.out.println( ballVolume );
```

Print the area of the ball
Print the volume of the ball
You may combine declaration and assignment in
<type> <name> = <value>;

```java
// compute values given a radius
public class RadiusAlt {
    public static void main( String[] args ) {
        //--- set the values of pi and radius
        double pi = 3.14159265;
        int radius = 10;

        // calculate the values
        double circlePerimeter = 2 * pi * radius;
        double circleArea = pi * radius * radius;
        double ballArea = 4 * pi * radius * radius;
        double ballVolume = 4 * pi * radius * radius * radius / 3;

        //-- output the values
    }
}
```
Computing the BMI

**Body-Mass Index** is given by the formula

\[
\text{BMI} = 703 \times \text{weight (in pounds)} / \text{height}^2 \text{ (in inches)}
\]

We consider the problem of computing BMI given a weight value and a height value
Computing the BMI

**Body-Mass Index** is given by the formula

\[
\text{BMI} = \frac{703 \times \text{weight (in pounds)}}{\text{height}^2 \text{ (in inches)}}
\]

We consider the problem of computing BMI given a weight value and a height value

- Declare variables for weight, height, and BMI
- Assign values to weight and height
- Compute the BMI value
- Print the result
Computing the BMI

**Body-Mass Index** is given by the formula

$$\text{BMI} = 703 \times \text{weight (in pounds)} / \text{height}^2 \text{ (in inches)}$$

We consider the problem of computing BMI given a weight value and a height value

- Declare variables for weight, height, and BMI
- Assign values to weight and height
- Compute the BMI value
- Print the result

Do the above twice
The Program

```java
public class BMI {
    public static void main( String[] args ) {
        double weight, height, bmi;
        // first time
        weight = 140.0;    // weight
        height = 67.0;     // height
        bmi = 703.0 * weight / (height * height);
        System.out.print( "weight = " );
        System.out.println( weight );
        System.out.print( "height = " );
        System.out.println( height );
        System.out.print( "BMI = " );
        System.out.println( bmi );
    }
}
```

Variable declarations
The Program

```java
public class BMI {
    public static void main( String[] args ) {
        double weight, height, bmi;
        // first time
        weight = 140.0; // weight
        height = 67.0; // height
        bmi = 703.0 * weight / (height * height);
        System.out.print( "weight = " );
        System.out.println( weight );
        System.out.print( "height = " );
        System.out.println( height );
        System.out.print( "BMI = " );
        System.out.println( bmi );
    }
}
```

Assignments (first round)
public class BMI {
    public static void main( String[] args ) {
        double weight, height, bmi;
        // first time
        weight = 140.0; // weight
        height = 67.0; // height
        bmi = 703.0 * weight / (height * height);
        System.out.print( "weight = " );
        System.out.println( weight );
        System.out.print( "height = " );
        System.out.println( height );
        System.out.print( "BMI = " );
        System.out.println( bmi );
    }
}

Calculation (first round)
The Program

```java
public class BMI {
    public static void main( String[] args ) {
        double weight, height, bmi;
        // first time
        weight = 140.0;    // weight
        height = 67.0;     // height
        bmi = 703.0 * weight / (height * height);
        System.out.print( "weight = " );
        System.out.println( weight );
        System.out.print( "height = " );
        System.out.println( height );
        System.out.print( "BMI = " );
        System.out.println( bmi );
    }
}
```

Printing the value of weight
The Program

```java
public class BMI {
    public static void main( String[] args ) {
        double weight, height, bmi;
        // first time
        weight = 140.0; // weight
        height = 67.0; // height
        bmi = 703.0 * weight / (height * height);
        System.out.print( "weight = " );
        System.out.println( weight );
        System.out.print( "height = " );
        System.out.println( height );
        System.out.print( "BMI = " );
        System.out.println( bmi );
    }
}
```

Printing the value of height
The Program

```java
public class BMI {
    public static void main( String[] args ) {
        double weight, height, bmi;
        // first time
        weight = 140.0;  // weight
        height = 67.0;   // height
        bmi = 703.0 * weight / (height * height);
        System.out.print( "weight = " );
        System.out.println( weight );
        System.out.print( "height = " );
        System.out.println( height );
        System.out.print( "BMI = " );
        System.out.println( bmi );
    }
}
```

Printing the value of bmi
The Program (cont’d)

// second time
weight = 150.0; // weight
height = 70.0; // height
bmi = 703.0 * weight / ( height * height );
System.out.print( "weight = " );
System.out.println( weight );
System.out.print( "height = " );
System.out.println( height );
System.out.print( "BMI = " );
System.out.println( bmi );
}
The Program (cont’d)

```java
// second time
weight = 150.0; // weight
height = 70.0; // height
bmi = 703.0 * weight / ( height * height );
System.out.print( "weight = " );
System.out.println( weight );
System.out.print( "height = " );
System.out.println( height );
System.out.print( "BMI = " );
System.out.println( bmi );
```

Printing the results
Table of Contents

1. Data, Literals, and Variables
2. Using Variables
3. Using Variables for Computation
4. Binary Operations on Numbers
Mathematical Short-hand

It is possible to short-hand expressions for updating a variable’s value with one operation.

Given an expression \( a = a \circ b; \) such that

- \( a \) is a number variable, \( \circ \) is one of \{\(+, −, /, \ast, \%\}\}, and \( b \) is an expression that produces a number, or
- \( a \) is a String variable, \( \circ \) is \(+\), and \( b \) is an expression,

we may write \( a \circ = b; \)

For example, we can write \( x += 3 \) in place of \( x = x + 3 \)
++ and --

For all number variables $x$,

- $x = x + 1$; can be simplified as $++x$; and as $x++$;
- $x = x - 1$; can be simplified as $--x$; and as $x--$;
++ and −−

For all number variables x,
- x = x + 1; can be simplified as as ++x; and as x++;
- x = x − 1; can be simplified as as −−x; and as x--; 

The ++ and −− can be attached to a variable appearing in a formula
- In ++x and −−x, the +1 and −1 to x occur before the evaluation of the formula
- In x++ and x--, the +1 and −1 to x occur after the evaluation of the formula
Combining Math Short-hand Expressions

```java
public class ShortHandExperiment {
    public static void main( String[] args ) {
        int myInt, other;
        myInt = 10;
        other = 13;
        System.out.print( "myInt is " );
        System.out.print( myInt );
        System.out.print( ", other is " );
        System.out.println( other );

        myInt += other;
        System.out.print( "Executed myInt += other	myInt is " );
        System.out.println( myInt );

        myInt *= other;
        System.out.print( "Executed myInt *= other	myInt is " );
        System.out.println( myInt );

        myInt -= other;
        System.out.print( "Executed myInt -= other	myInt is " );
        System.out.println( myInt );
    }
}
```

Variable declaration
Combining Math Short-hand Expressions

```java
public class ShortHandExperiment {
    public static void main( String[] args ) {
        int myInt, other;
        myInt = 10;
        other = 13;
        System.out.print( "myInt is " );
        System.out.print( myInt );
        System.out.print( ", other is " );
        System.out.println( other );
        myInt += other;
        System.out.print( "Executed myInt += other	myInt is " );
        System.out.println( myInt );
        myInt *= other;
        System.out.print( "Executed myInt *= other	myInt is " );
        System.out.println( myInt );
        myInt -= other;
        System.out.print( "Executed myInt -= other	myInt is " );
        System.out.println( myInt );
    }
}
```

Initial assignments
Combining Math Short-hand Expressions

```java
public class ShortHandExperiment {
    public static void main( String[] args ) {
        int myInt, other;
        myInt = 10;
        other = 13;

        System.out.print( "myInt is " );
        System.out.print( myInt );
        System.out.print( "other is " );
        System.out.println( other );

        myInt += other;
        System.out.print( "Executed myInt += other	myInt is " );
        System.out.println( myInt );

        myInt *= other;
        System.out.print( "Executed myInt *= other	myInt is " );
        System.out.println( myInt );

        myInt -= other;
        System.out.print( "Executed myInt -= other	myInt is " );
        System.out.println( myInt );
    }
}
```

Print the "myInt is " followed by the value of myInt
Combining Math Short-hand Expressions

```java
public class ShortHandExperiment {
    public static void main( String[] args ) {
        int myInt, other;
        myInt = 10;
        other = 13;
        System.out.print( "myInt is " );
        System.out.print( myInt );
        System.out.print(", other is " );
        System.out.println( other );

        myInt += other;
        System.out.print( "Executed myInt += other\tmyInt is " );
        System.out.println( myInt );

        myInt *= other;
        System.out.print( "Executed myInt *= other\tmyInt is " );
        System.out.println( myInt );

        myInt -= other;
        System.out.print( "Executed myInt -= other\tmyInt is " );
        System.out.println( myInt );
    }
}
```

Continued with ", other is " followed by the value of other
Combining Math Short-hand Expressions

```java
public class ShortHandExperiment {
    public static void main( String[] args ) {
        int myInt, other;
        myInt = 10;
        other = 13;
        System.out.print( "myInt is " );
        System.out.print( myInt );
        System.out.print( ", other is " );
        System.out.println( other );

        myInt += other;
        System.out.print( "Executed myInt += other\tmyInt is " );
        System.out.println( myInt );

        myInt *= other;
        System.out.print( "Executed myInt *= other\tmyInt is " );
        System.out.println( myInt );

        myInt -= other;
        System.out.print( "Executed myInt -= other\tmyInt is " );
        System.out.println( myInt );
    }
}
```

Perform `myInt += other` and report the outcome.
Combining Math Short-hand Expressions

```java
public class ShortHandExperiment {
    public static void main( String[] args ) {
        int myInt, other;
        myInt = 10;
        other = 13;
        System.out.print( "myInt is " );
        System.out.print( myInt );
        System.out.print( ", other is " );
        System.out.println( other );

        myInt += other;
        System.out.print( "Executed myInt += other	myInt is " );
        System.out.println( myInt );

        myInt *= other;
        System.out.print( "Executed myInt *= other	myInt is " );
        System.out.println( myInt );

        myInt -= other;
        System.out.print( "Executed myInt -= other	myInt is " );
        System.out.println( myInt );
    }
}
```

Perform \texttt{myInt *= other} and report the outcome.
Combining Math Short-hand Expressions

```java
public class ShortHandExperiment {
    public static void main( String[] args ) {
        int myInt, other;
        myInt = 10;
        other = 13;
        System.out.print( "myInt is " );
        System.out.print( myInt );
        System.out.print( ", other is " );
        System.out.println( other );

        myInt += other;
        System.out.print( "Executed myInt += other\tmyInt is " );
        System.out.println( myInt );

        myInt *= other;
        System.out.print( "Executed myInt *= other\tmyInt is " );
        System.out.println( myInt );

        myInt -= other;
        System.out.print( "Executed myInt -= other\tmyInt is " );
        System.out.println( myInt );
    }
}
```

Perform `myInt -= other` and report the outcome
Combining Math Short-hand Expressions

```java
myInt /= other;
System.out.print( "Executed myInt /= other\tmyInt is " );
System.out.println( myInt );

myInt %= other;
System.out.print( "Executed myInt %= other\tmyInt is " );
System.out.println( myInt );
```

Perform `myInt /= other` and report the outcome
Combining Math Short-hand Expressions

```java
myInt /= other;
System.out.print( "Executed myInt /= other\tmyInt is " );
System.out.println( myInt );

myInt %= other;
System.out.print( "Executed myInt %= other\tmyInt is " );
System.out.println( myInt );
```

Perform `myInt %= other` and report the outcome
Combining Math Short-hand Expressions

```java
myInt += ++other;
System.out.print( "Executed myInt += ++other\tmyInt is " );
System.out.print( myInt );
System.out.print( ", other is now " );
System.out.println( other );

myInt += other++;
System.out.print( "Executed myInt += other++\tmyInt is " );
System.out.print( myInt );
System.out.print( ", other is now " );
System.out.println( other );

myInt += --other;
System.out.print( "Executed myInt += --other\tmyInt is " );
System.out.print( myInt );
System.out.print( ", other is now " );
System.out.println( other );

myInt += other--;
System.out.print( "Executed myInt += other--\tmyInt is " );
System.out.print( myInt );
System.out.print( ", other is now " );
System.out.println( other );
```

Perform `myInt += ++other`
Combining Math Short-hand Expressions

```java
test = 10;
myInt += ++other;
System.out.print( "Executed myInt += ++other\tmyInt is " );
System.out.print( myInt );
System.out.print( ", other is now " );
System.out.println( other );

myInt += other++;
System.out.print( "Executed myInt += other++\tmyInt is " );
System.out.print( myInt );
System.out.print( ", other is now " );
System.out.println( other );

myInt += --other;
System.out.print( "Executed myInt += --other\tmyInt is " );
System.out.print( myInt );
System.out.print( ", other is now " );
System.out.println( other );

myInt += other--;```

Print the values
Combining Math Short-hand Expressions

```java
myInt += ++other;
System.out.print( "Executed myInt += ++other\nmyInt is " );
System.out.print( myInt );
System.out.print( ", other is now " );
System.out.println( other );

myInt += other++;    // Error: Incorrect use of the short-hand increment operator
System.out.print( "Executed myInt += other++\nmyInt is " );
System.out.print( myInt );
System.out.print( ", other is now " );
System.out.println( other );

myInt += --other;
System.out.print( "Executed myInt += --other\nmyInt is " );
System.out.print( myInt );
System.out.print( ", other is now " );
System.out.println( other );

myInt += other--;  // Error: Incorrect use of the short-hand decrement operator
System.out.print( "Executed myInt += other--\nmyInt is " );
System.out.print( myInt );
System.out.print( ", other is now " );
System.out.println( other );
```

Do the same with `myInt += other++`
Combining Math Short-hand Expressions

```java
myInt += ++other;
System.out.print( "Executed myInt += ++other\tmyInt is " );
System.out.print( myInt );
System.out.print( "\t, other is now " );
System.out.println( other );

myInt += other++;  
System.out.print( "Executed myInt += other++\tmyInt is " );
System.out.print( myInt );
System.out.print( "\t, other is now " );
System.out.println( other );

myInt += --other;
System.out.print( "Executed myInt += --other\tmyInt is " );
System.out.print( myInt );
System.out.print( "\t, other is now " );
System.out.println( other );

myInt += other--;  
System.out.print( "Executed myInt += other--\tmyInt is " );
System.out.print( myInt );
System.out.print( "\t, other is now " );
System.out.println( other );
```

Do the same with `myInt += -other`
Combining Math Short-hand Expressions

```java
myInt += ++other;
System.out.print("Executed myInt += ++other\tmyInt is ");
System.out.print(myInt);
System.out.print( ", other is now ");
System.out.println( other );

myInt += other++;
System.out.print("Executed myInt += other++\tmyInt is ");
System.out.print(myInt);
System.out.print( ", other is now ");
System.out.println( other );

myInt += --other;
System.out.print("Executed myInt += --other\tmyInt is ");
System.out.print(myInt);
System.out.print( ", other is now ");
System.out.println( other );

myInt += other--;  // Do the same with myInt += other--
System.out.print("Executed myInt += other--\tmyInt is ");
System.out.print(myInt);
System.out.print( ", other is now ");
System.out.println( other );
```

Type Requirement

You cannot store a real number value to an int variable

```java
int result = 3.9 * 4.5;
```

will produce a compilation error

To covert, you can truncate the real value using a prefix of `(int)`

```java
int result = (int)( 3.9 * 4.5 );
```

will assign the vale of 17 (since 3.9 * 4.5 = 17.55) to `result`