

Introduction to Java: Expressions and Variables

Lecture 4

CS106A, Summer 2019

Sarai Gould & Laura Cruz-Albrecht

With inspiration from slides created by Keith Schwarz, Mehran Sahami, Eric Roberts, Stuart Reges, Chris Piech and others.



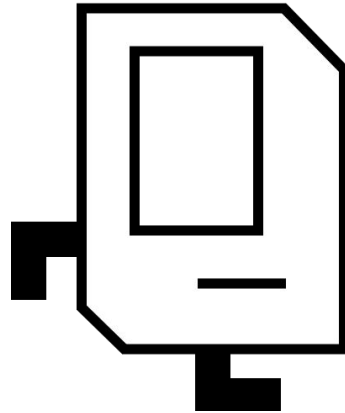
Announcements

- If you have midterm conflicts, email both instructors ASAP **and** fill out the following form (by July 8th at the latest):
 - <http://bit.ly/CS106AMidtermConflicts>
- Please email both instructors as soon as possible if you have academic accommodations from the OAE
- Assignment 1 out!
- “Extra” slides: treat these as required reading

Plan for Today

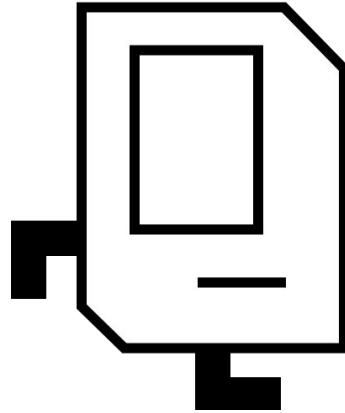
- Bye, Karel!
- Console programs
- Variables
- Expressions
- Practice: Receipt

Bye, Karel!



I will miss u

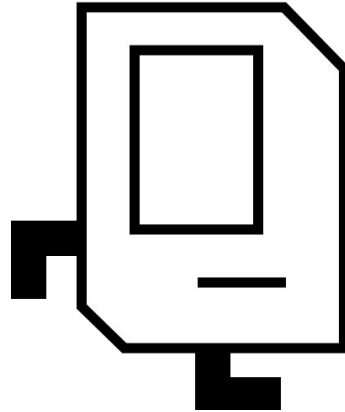
Bye, Karel!



I will miss u

Enjoy Java! 

Bye, Karel!

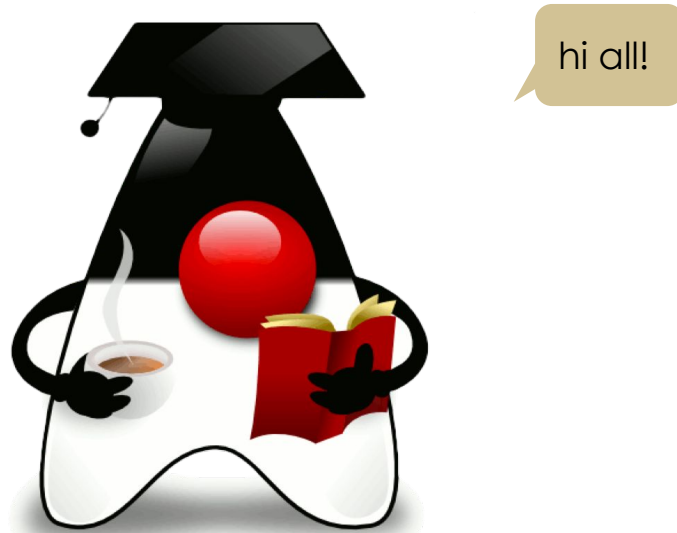


I will miss u

Enjoy Java! ☕

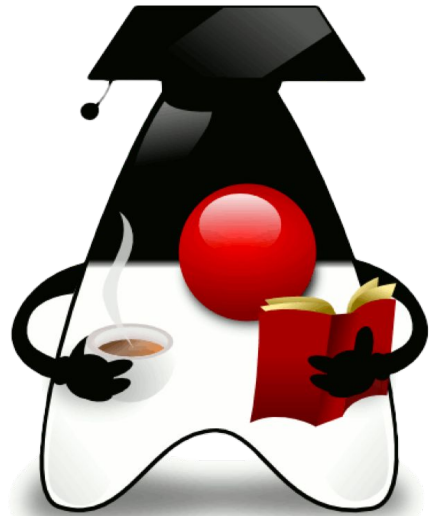
See you on the
midterm 😬

Hello, Java!



* fun fact: this is the official Java mascot

Hello, Java!

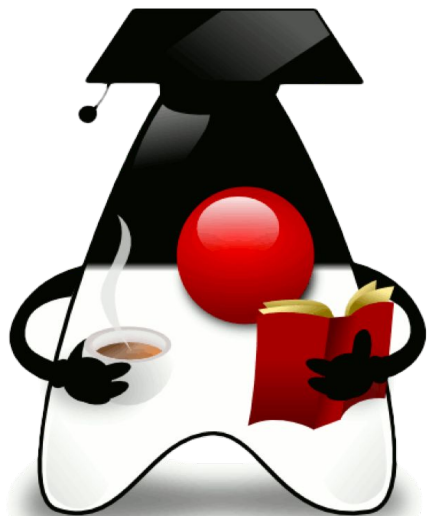


hi all!

I like Karel a latte ❤️
... but i'm cooler

* fun fact: this is the official Java mascot

Hello, Java!



hi all!

I like Karel a latte ❤️
... but i'm cooler

Can't start your day
without me ☕

* fun fact: this is the official Java mascot

Plan for Today

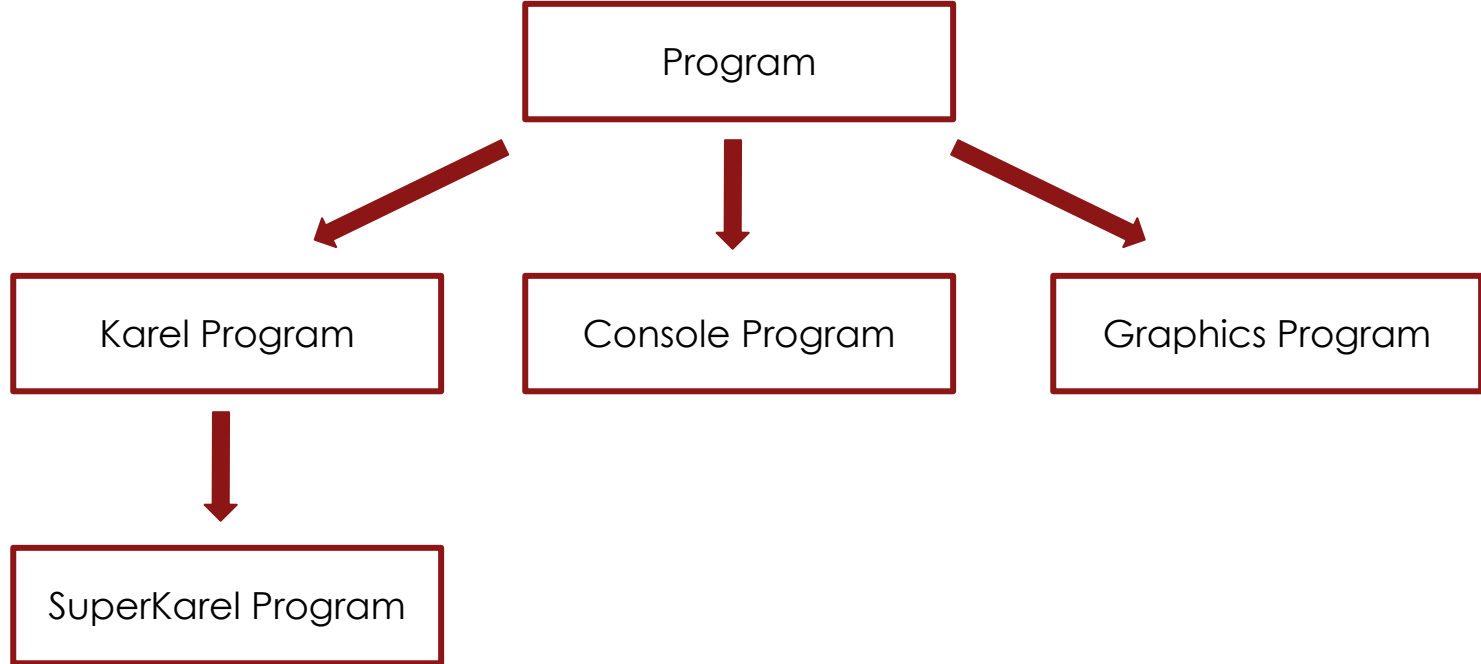
- Bye, Karel!
- Console programs
- Variables
- Expressions
- Practice: Receipt

Types of Programs

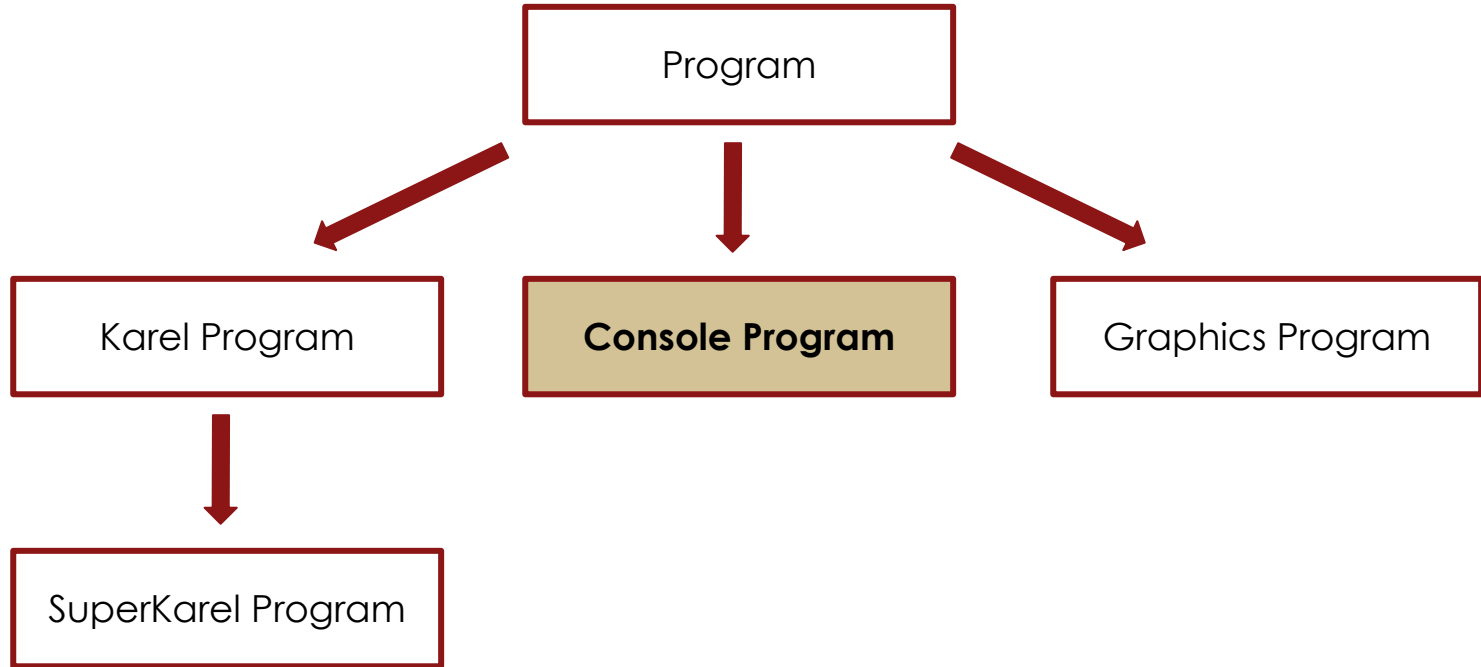


Program

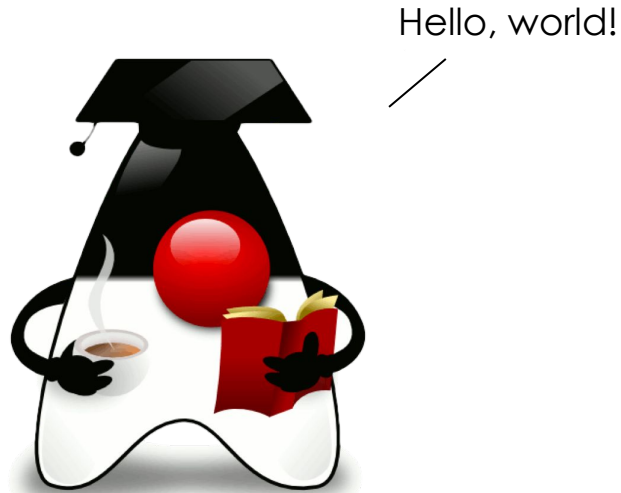
Types of Programs



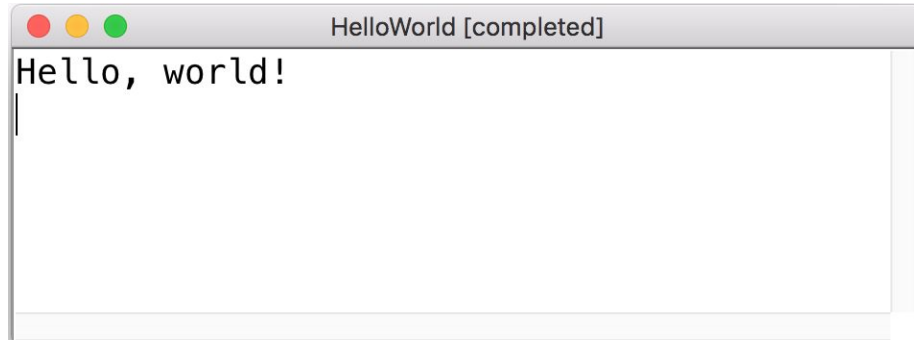
Types of Programs



Our First Java Program



Hello, World

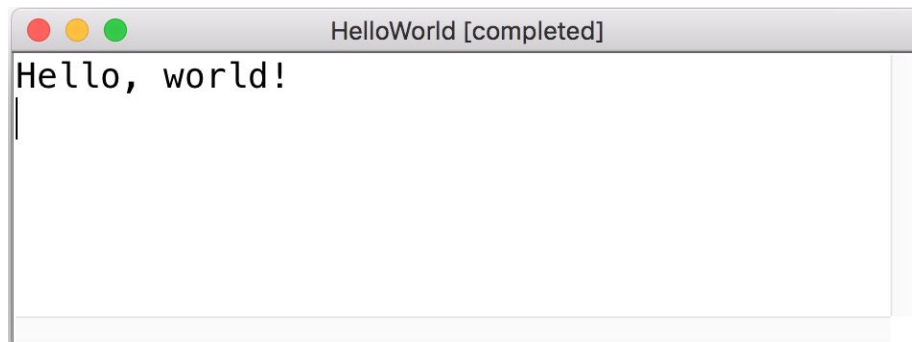


```
HelloWorld [completed]
Hello, world!
```

Our First Java Program

```
import acm.program.*;

public class HelloWorld extends ConsoleProgram {
    public void run() {
        println("Hello, world!");
    }
}
```



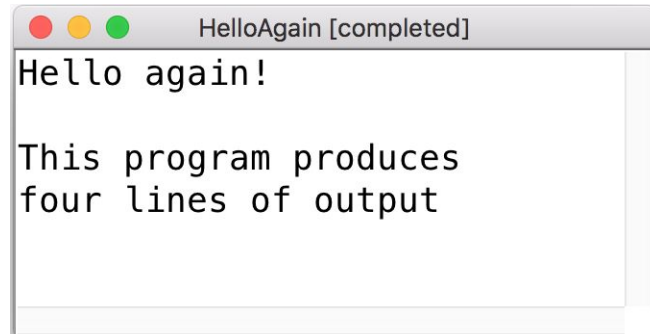
Console Programs

```
import acm.program.*;
public class Name extends ConsoleProgram {
    public void run() {
        statements;
    }
}
```

- Unlike Karel, many programs produce their behavior as text
- Console: textbox into which the behavior is displayed
 - *Output*: messages displayed by the program
 - *Input*: data read by the program that the user types

Console Programs

```
public class HelloAgain extends ConsoleProgram {  
    public void run() {  
        println("Hello again!");  
        println();  
        println("This program produces");  
        println("four lines of output");  
    }  
}
```



Console Output: println

- `println`: a statement that prints a line of *output* on the console, and goes to the next line
- Two uses:

```
println("TEXT");    // prints the string TEXT  
println();          // prints blank line
```

Console I/O

- I/O: Input/Output
- `println` allows out to output text to the user via the console
- We can also get input from the user via the console
 - But before we can get input from the user, we need a way to store it...

Plan for Today

- Bye, Karel!
- Console programs
- **Variables**
- Expressions
- Practice: Receipt

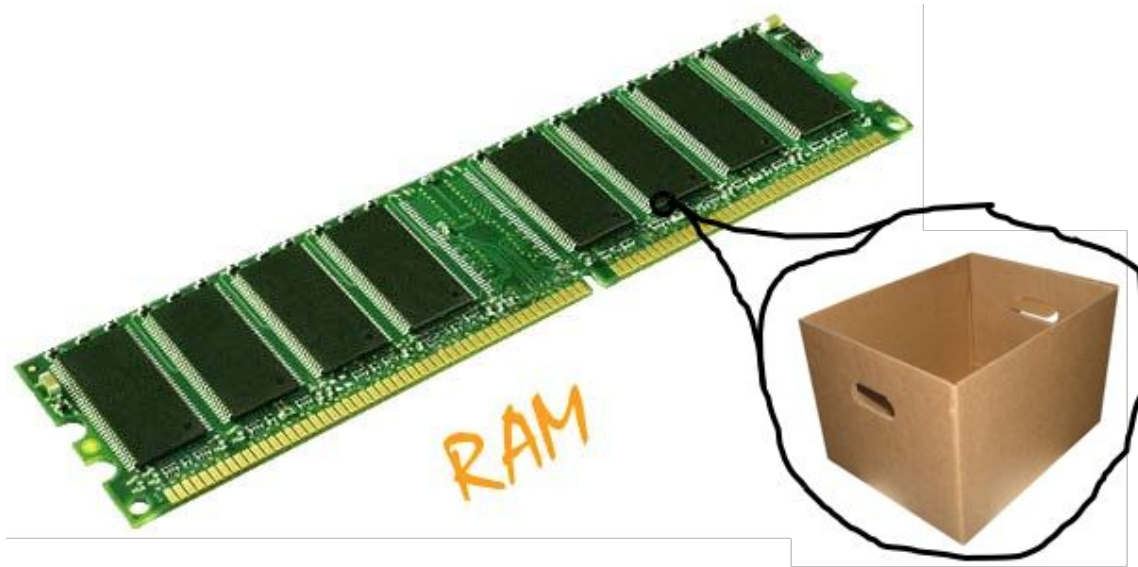


What's a variable?

Variables = Boxes



Variables = Boxes



* my computer has space for about 2 billion boxes

Making a New Variable

```
int myVar = 22;
```

Making a New Variable

```
int myVar = 22;
```

(contains an `int`)



Making a New Variable

```
int myVar = 22;
```

(contains an `int`)



Making a New Variable

```
int myVar = 22;
```

(contains an `int`)



3 Properties

type
↓
int myVar = 22;

type
↓
(contains an **int**)



3 Properties

type name
↓ ↓
int **myVar** = 22;



3 Properties

type name value
↓ ↓ ↓
int **myVar** = **22;**



Variable Types

```
// an integer number
```

```
int num = 5;
```

```
// a real (decimal) number
```

```
double fraction = 0.2;
```

```
// true or false
```

```
boolean isSummer = true;
```


Variable Types

```
// a single character
```

```
char letter = 'c';
```

```
// a “string” of text
```

```
String phrase = “Hi!”;
```

```
String alsoAString = “5”; // not an int!
```

Double vs. Int

Double - answer is a decimal
How much do I weigh?



Int - answer is an integer
How many pets do I have?



Making a Variable

```
int myVar = 22;
```

Making a Variable

```
int myVar;  
myVar = 22;
```

Making a Variable

```
int myVar; // declare  
myVar = 22; // assign
```

Declaration & Assignment

- **Declaration:** sets aside memory for storing a value
 - Variables must be declared before they can be used

Declaration & Assignment

- **Declaration:** sets aside memory for storing a value
 - Variables must be declared before they can be used

```
int myBox;
```



myBox

Declaration & Assignment

- **Declaration:** sets aside memory for storing a value
 - Variables must be declared before they can be used

```
int myBox;
```



myBox

- **Assignment:** stores a value into an **existing** variable
 - value can be an expression; variable stores its result

Declaration & Assignment

- **Declaration:** sets aside memory for storing a value
 - Variables must be declared before they can be used

```
int myBox;
```



myBox

- **Assignment:** stores a value into an **existing** variable
 - value can be an expression; variable stores its result

```
double cost;           // declaration
```



cost

Declaration & Assignment

- **Declaration:** sets aside memory for storing a value
 - Variables must be declared before they can be used

```
int myBox;
```



myBox

- **Assignment:** stores a value into an **existing** variable
 - value can be an expression; variable stores its result

```
double cost;           // declaration  
cost = 1.5 + 0.75;    // assignment
```



cost

Declaration & Assignment

- **Declaration:** sets aside memory for storing a value
 - Variables must be declared before they can be used

```
int myBox;
```



myBox

- **Assignment:** stores a value into an **existing** variable
 - value can be an expression; variable stores its result

```
double cost;           // declaration  
cost = 1.5 + 0.75;    // assignment
```



cost

- Commonly, a variable is declared + initialized in one statement

Declaration & Assignment

- **Declaration:** sets aside memory for storing a value
 - Variables must be declared before they can be used

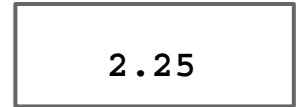
```
int myBox;
```



myBox

- **Assignment:** stores a value into an **existing** variable
 - value can be an expression; variable stores its result

```
double cost; // declaration  
cost = 1.5 + 0.75; // assignment
```



cost

- Commonly, a variable is declared + initialized in one statement

```
int saraisBox = 4;
```



saraisBox

What Happens Here?

- What happens here?

```
int x = 3;
```

```
x = x + 2; // ?
```



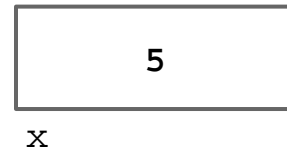
x

What Happens Here?

- What happens here?

```
int x = 3;
```

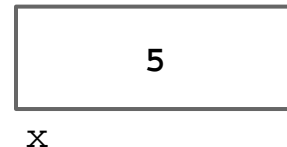
```
x = x + 2; // 5
```



'=' Means Assignment

- What happens here?

```
int x = 3;  
x = x + 2; // 5
```



- Assignment uses =, but it's not an algebraic equation
 - = means: *store the value at right in the variable at left*
 - Right side expression evaluated first, then stored in variable at left

Outputting Variable Value

```
// creates variable called temperature with value 64.8  
double temperature = (72.2 + 57.4) / 2.0;
```


Outputting Variable Value

```
// creates variable called temperature with value 64.8  
double temperature = (72.2 + 57.4) / 2.0;
```

```
// prints value of temperature variable to the console  
println(temperature);    // 64.8
```

Outputting Variable Value

```
// creates variable called temperature with value 64.8  
double temperature = (72.2 + 57.4) / 2.0;
```

```
// prints value of temperature variable to the console  
println(temperature);    // 64.8
```

```
// can use + to print string and variable's value  
println("avg temp is " + temperature); // avg temp is 64.8
```

Using Variables

- Once given a value, variables can be used in expressions

```
int myVar = 22;  
println(2 * myVar + 1); // 45
```

Using Variables

- You can assign a value more than once

Using Variables

- You can assign a value more than once

64.8

temperature

```
// declare  
double temperature = (72.2 + 57.4) / 2.0;  
println("temp yesterday: " + temperature); // temp yesterday: 64.8
```

Using Variables

- You can assign a value more than once

72.1

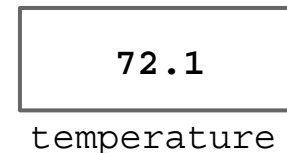
temperature

```
// declare
double temperature = (72.2 + 57.4) / 2.0;
println("temp yesterday: " + temperature); // temp yesterday: 64.8

// reassign
temperature = 72.1;
```

Using Variables

- You can assign a value more than once



```
// declare
double temperature = (72.2 + 57.4) / 2.0;
println("temp yesterday: " + temperature); // temp yesterday: 64.8

// reassign
temperature = 72.1;
println("temp today: " + temperature); // temp today: 72.1
```

Assignment and Types

- A variable can only store a value of its own type

```
int x = 2.5; // Error: incompatible types
```



Assignment and Types

- A variable can only store a value of its own type

```
int x = 2.5; // Error: incompatible types
```



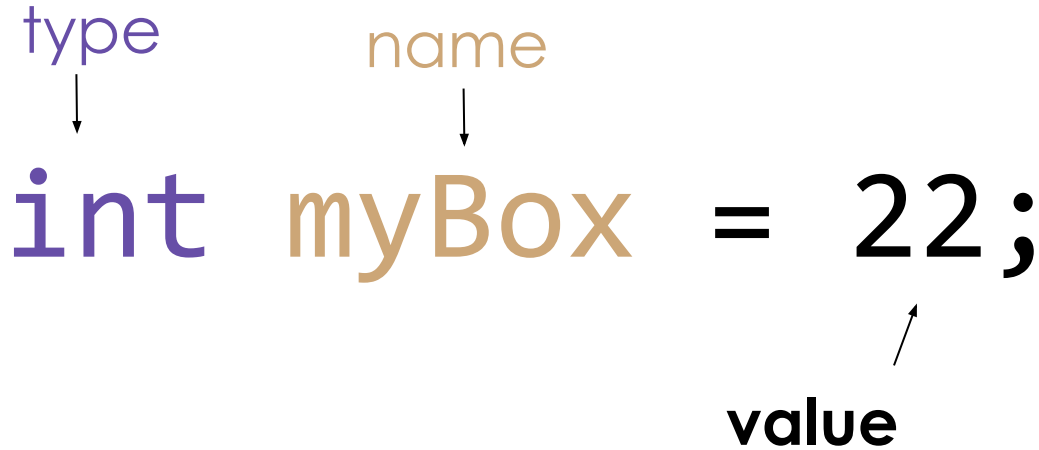
- An int value can be stored in a double variable
 - The value is converted into the equivalent real number

```
double gpa = 4;
```

4.0

gpa

Setting Variable Values



The diagram illustrates the components of a variable declaration. It shows the code `int myBox = 22;` with three labels and arrows pointing to their respective parts:

- The label `type` (in purple) has a downward arrow pointing to the word `int`.
- The label `name` (in orange) has a downward arrow pointing to the word `myBox`.
- The label `value` (in black) has an upward arrow pointing to the number `22`.

```
int myBox = 22;
```

Setting Variable Values

type name

↓ ↓

int **myBox** = *expression*;

 ↑

 value

The diagram illustrates the components of a variable declaration. The word 'type' is shown in purple above the code 'int', with a downward arrow pointing to it. The word 'name' is shown in orange above 'myBox', with a downward arrow pointing to it. The word 'value' is shown in black below 'expression', with an upward arrow pointing to it. The code 'int myBox = expression;' is written in a mix of bold black, purple, orange, and italicized black fonts.

Setting Variable Values

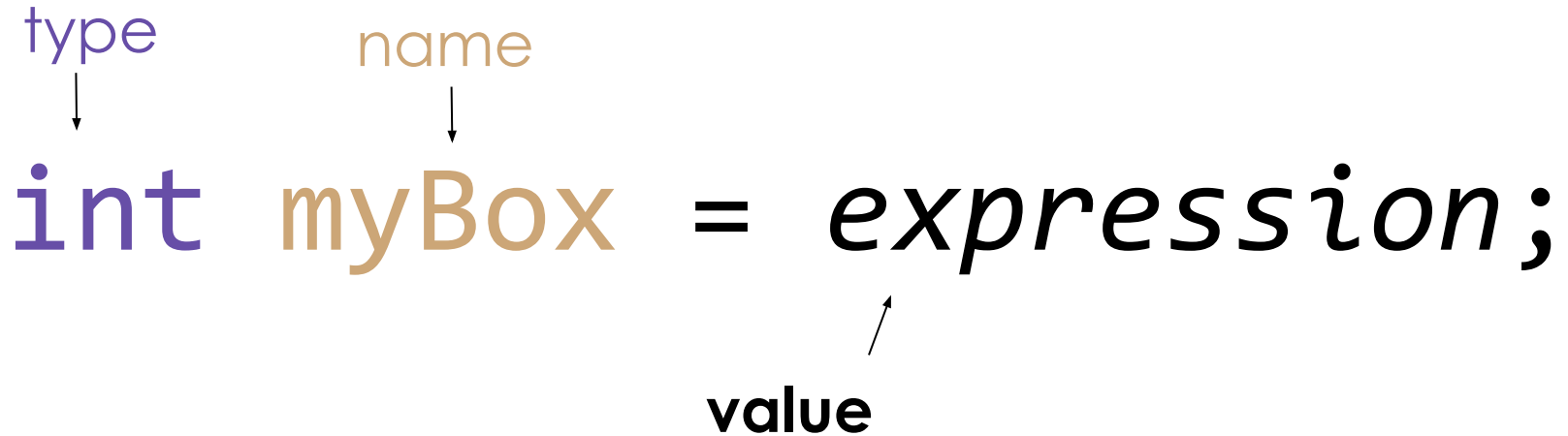
type name

↓ ↓

int **myBox** = *expression* ;

 ↑

value



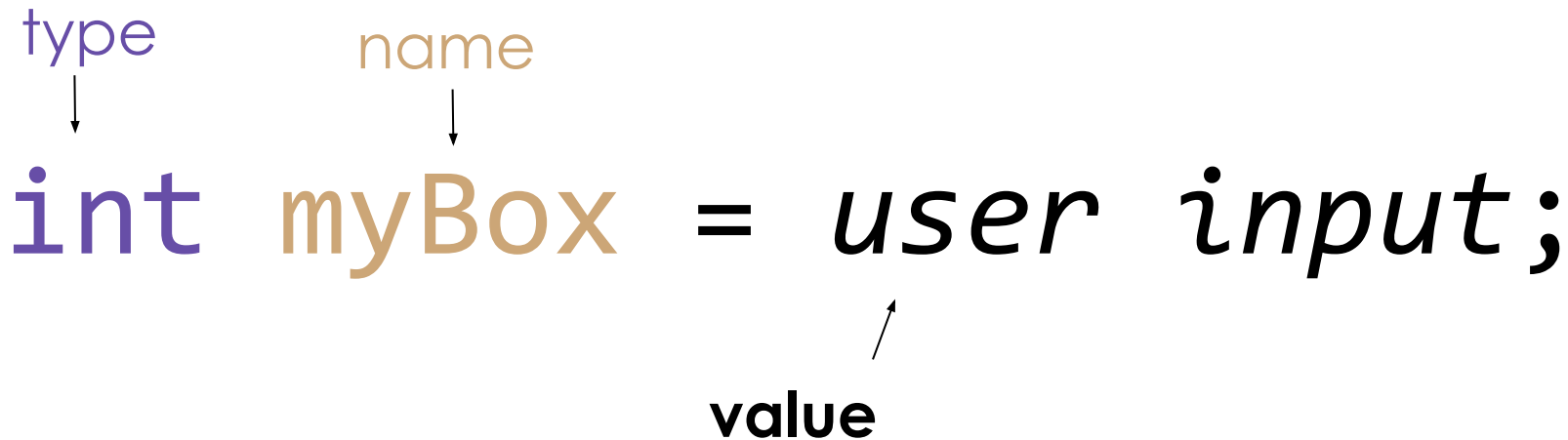
example:

```
double temperature = (72.2 + 57.4) / 2.0; // 64.8
```



Let's now get values from
user input!

Setting Variable Values



The diagram shows the code `int myBox = user input;` with three annotations. The word `int` is in purple, `myBox` is in brown, and `user input;` is in black. An arrow labeled 'type' points to `int`. An arrow labeled 'name' points to `myBox`. An arrow labeled 'value' points to `user input;`.

```
type  
↓  
int myBox = user input;  
name  
↓  
value  
↗
```

Values from User Input

```
// Prompts user for a whole number. Stores result  
// in a variable (aka a box).
```

```
int pets = readInt("How many pets? ");
```

```
// Prompts user for a decimal number. Stores result  
// in a variable (aka a box).
```

```
double tip = readDouble("Tip? $ ");
```

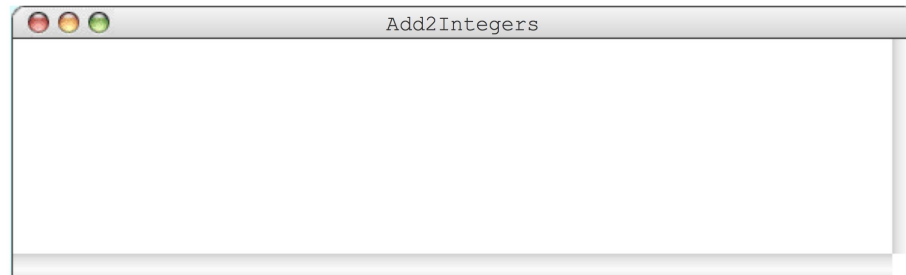
```
// Prompts user for a boolean & stores result in variable.
```

```
boolean isSunny = readBoolean("Sun shining? ");
```

Add2Integers

```
public class Add2Integers extends ConsoleProgram {  
    public void run() {  
        println("This program adds two numbers.");  
        int n1 = readInt("Enter n1: ");  
        int n2 = readInt("Enter n2: ");  
        int total = n1 + n2;  
        println("The total is " + total + ".");  
    }  
}
```

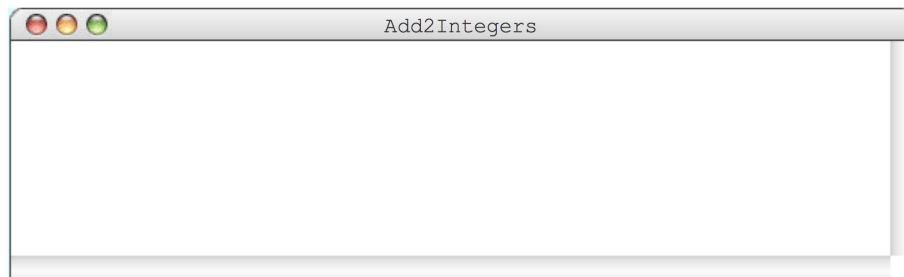
n1	n2	total
<input type="text"/>	<input type="text"/>	<input type="text"/>



Add2Integers

```
public class Add2Integers extends ConsoleProgram {  
    public void run() {  
        println("This program adds two numbers.");  
        int n1 = readInt("Enter n1: ");  
        int n2 = readInt("Enter n2: ");  
        int total = n1 + n2;  
        println("The total is " + total + ".");  
    }  
}
```

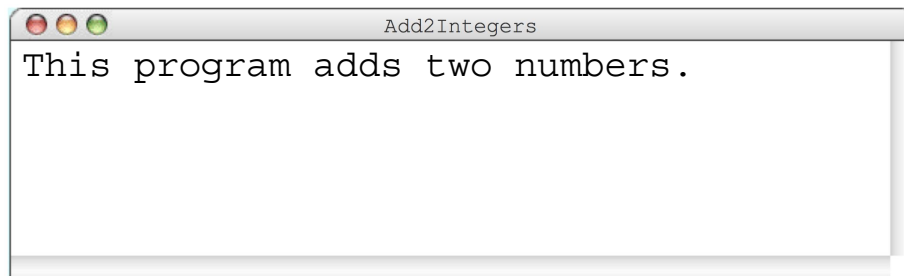
n1	n2	total
<input type="text"/>	<input type="text"/>	<input type="text"/>



Add2Integers

```
public class Add2Integers extends ConsoleProgram {  
    public void run() {  
        println("This program adds two numbers.");  
        int n1 = readInt("Enter n1: ");  
        int n2 = readInt("Enter n2: ");  
        int total = n1 + n2;  
        println("The total is " + total + ".");  
    }  
}
```

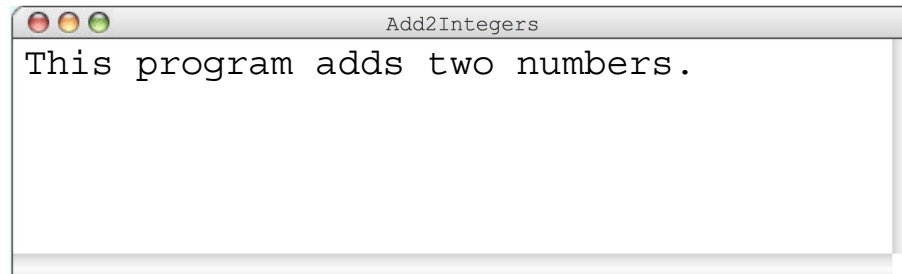
n1	n2	total
<input type="text"/>	<input type="text"/>	<input type="text"/>



Add2Integers

```
public class Add2Integers extends ConsoleProgram {  
    public void run() {  
        println("This program adds two numbers.");  
        int n1 = readInt("Enter n1: ");  
        int n2 = readInt("Enter n2: ");  
        int total = n1 + n2;  
        println("The total is " + total + ".");  
    }  
}
```

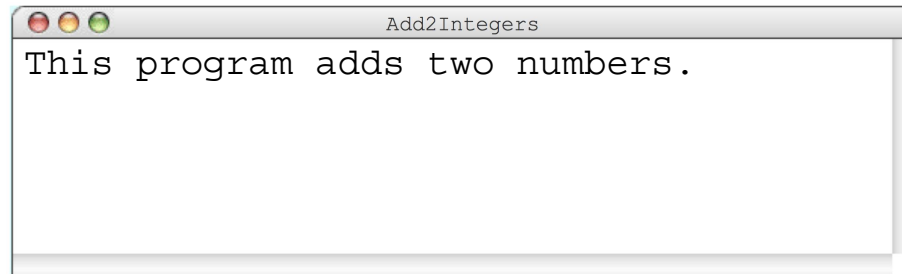
n1	n2	total
<input type="text"/>	<input type="text"/>	<input type="text"/>



Add2Integers

```
public class Add2Integers extends ConsoleProgram {  
    public void run() {  
        println("This program adds two numbers.");  
        int n1 = readInt("Enter n1: ");  
        int n2 = readInt("Enter n2: ");  
        int total = n1 + n2;  
        println("The total is " + total + ".");  
    }  
}
```

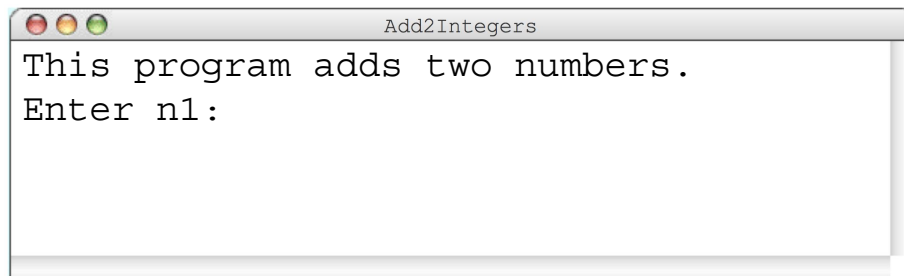
n1	n2	total
<input type="text"/>	<input type="text"/>	<input type="text"/>



Add2Integers

```
public class Add2Integers extends ConsoleProgram {  
    public void run() {  
        println("This program adds two numbers.");  
        int n1 = readInt("Enter n1: ");  
        int n2 = readInt("Enter n2: ");  
        int total = n1 + n2;  
        println("The total is " + total + ".");  
    }  
}
```

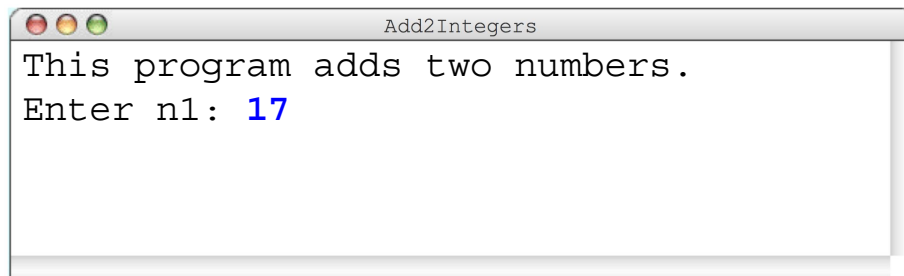
n1	n2	total
<input type="text"/>	<input type="text"/>	<input type="text"/>



Add2Integers

```
public class Add2Integers extends ConsoleProgram {  
    public void run() {  
        println("This program adds two numbers.");  
        int n1 = readInt("Enter n1: ");  
        int n2 = readInt("Enter n2: ");  
        int total = n1 + n2;  
        println("The total is " + total + ".");  
    }  
}
```

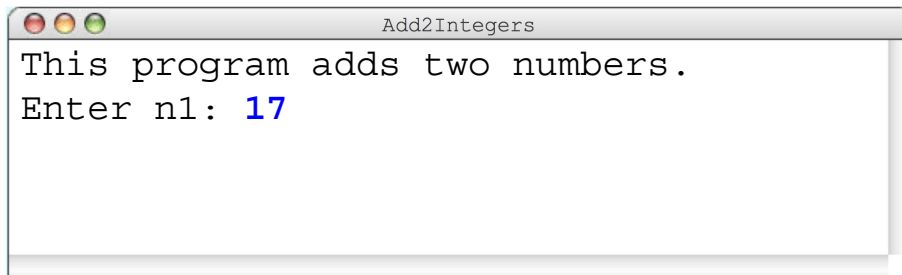
n1	n2	total
<input type="text"/>	<input type="text"/>	<input type="text"/>



Add2Integers

```
public class Add2Integers extends ConsoleProgram {  
    public void run() {  
        println("This program adds two numbers.");  
        int n1 = readInt("Enter n1: ");  
        int n2 = readInt("Enter n2: ");  
        int total = n1 + n2;  
        println("The total is " + total + ".");  
    }  
}
```

n1	n2	total
<input type="text" value="17"/>	<input type="text"/>	<input type="text"/>

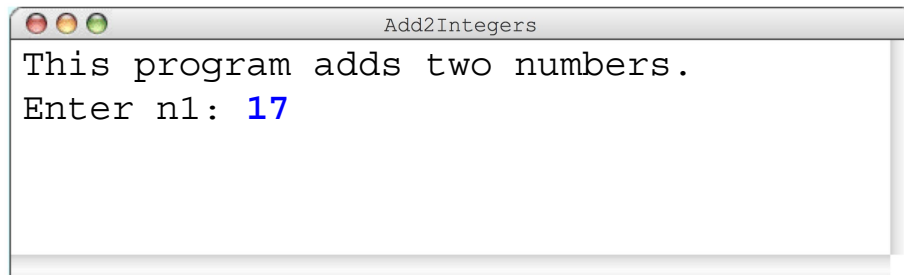


The screenshot shows a window titled "Add2Integers" with a white background and a grey title bar. The text inside the window reads: "This program adds two numbers." followed by "Enter n1: 17". The number "17" is highlighted in blue, indicating it was entered by the user.

Add2Integers

```
public class Add2Integers extends ConsoleProgram {  
    public void run() {  
        println("This program adds two numbers.");  
        int n1 = readInt("Enter n1: ");  
        int n2 = readInt("Enter n2: ");  
        int total = n1 + n2;  
        println("The total is " + total + ".");  
    }  
}
```

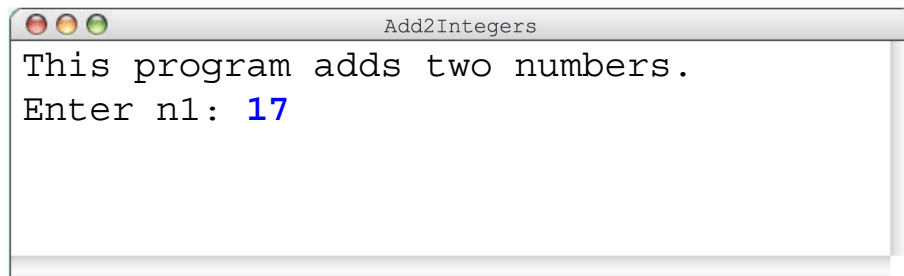
n1	n2	total
<input type="text" value="17"/>	<input type="text"/>	<input type="text"/>



Add2Integers

```
public class Add2Integers extends ConsoleProgram {  
    public void run() {  
        println("This program adds two numbers.");  
        int n1 = readInt("Enter n1: ");  
        int n2 = readInt("Enter n2: ");  
        int total = n1 + n2;  
        println("The total is " + total + ".");  
    }  
}
```

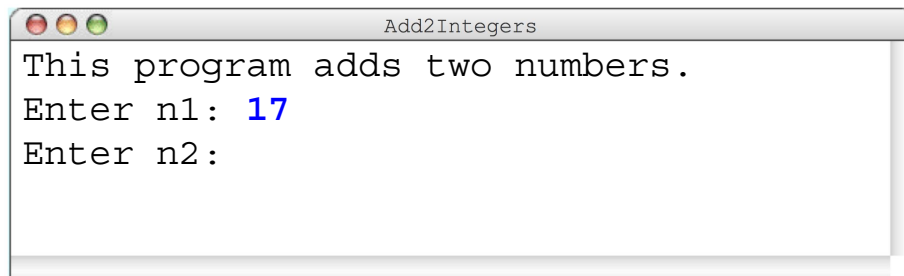
n1	n2	total
<input type="text" value="17"/>	<input type="text"/>	<input type="text"/>



Add2Integers

```
public class Add2Integers extends ConsoleProgram {  
    public void run() {  
        println("This program adds two numbers.");  
        int n1 = readInt("Enter n1: ");  
        int n2 = readInt("Enter n2: ");  
        int total = n1 + n2;  
        println("The total is " + total + ".");  
    }  
}
```

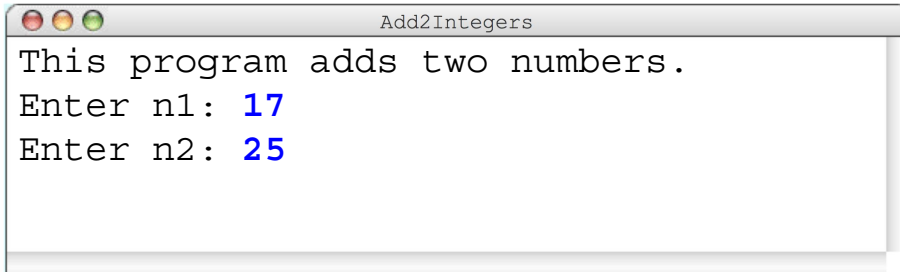
n1	n2	total
<input type="text" value="17"/>	<input type="text"/>	<input type="text"/>



Add2Integers

```
public class Add2Integers extends ConsoleProgram {  
    public void run() {  
        println("This program adds two numbers.");  
        int n1 = readInt("Enter n1: ");  
        int n2 = readInt("Enter n2: ");  
        int total = n1 + n2;  
        println("The total is " + total + ".");  
    }  
}
```

n1	n2	total
<input type="text" value="17"/>	<input type="text"/>	<input type="text"/>



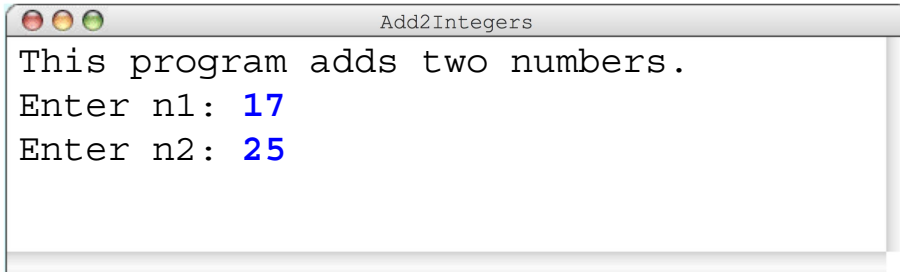
The screenshot shows a window titled "Add2Integers" with a white background and a grey title bar. The text inside the window is as follows:

```
This program adds two numbers.  
Enter n1: 17  
Enter n2: 25
```

Add2Integers

```
public class Add2Integers extends ConsoleProgram {  
    public void run() {  
        println("This program adds two numbers.");  
        int n1 = readInt("Enter n1: ");  
        int n2 = readInt("Enter n2: ");  
        int total = n1 + n2;  
        println("The total is " + total + ".");  
    }  
}
```

n1	n2	total
17	25	



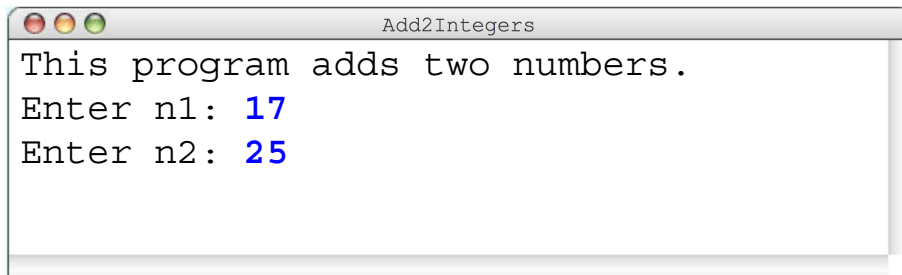
The screenshot shows a window titled "Add2Integers" with a white background and a grey title bar. The text inside the window is as follows:

```
This program adds two numbers.  
Enter n1: 17  
Enter n2: 25
```

Add2Integers

```
public class Add2Integers extends ConsoleProgram {  
    public void run() {  
        println("This program adds two numbers.");  
        int n1 = readInt("Enter n1: ");  
        int n2 = readInt("Enter n2: ");  
        int total = n1 + n2;  
        println("The total is " + total + ".");  
    }  
}
```

n1	n2	total
17	25	



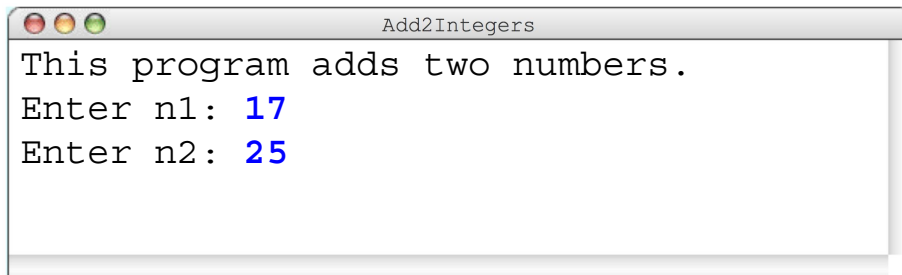
The screenshot shows a window titled "Add2Integers" with a white background and a grey title bar. The text inside the window is as follows:

```
This program adds two numbers.  
Enter n1: 17  
Enter n2: 25
```

Add2Integers

```
public class Add2Integers extends ConsoleProgram {  
    public void run() {  
        println("This program adds two numbers.");  
        int n1 = readInt("Enter n1: ");  
        int n2 = readInt("Enter n2: ");  
        int total = n1 + n2;  
        println("The total is " + total + ".");  
    }  
}
```

n1	n2	total
17	25	42



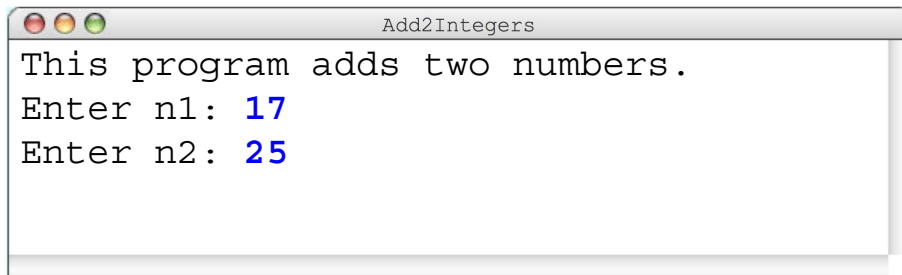
The screenshot shows a window titled "Add2Integers" with a white background and a grey title bar. The text inside the window is as follows:

```
This program adds two numbers.  
Enter n1: 17  
Enter n2: 25
```

Add2Integers

```
public class Add2Integers extends ConsoleProgram {  
    public void run() {  
        println("This program adds two numbers.");  
        int n1 = readInt("Enter n1: ");  
        int n2 = readInt("Enter n2: ");  
        int total = n1 + n2;  
        println("The total is " + total + ".");  
    }  
}
```

n1	n2	total
17	25	42



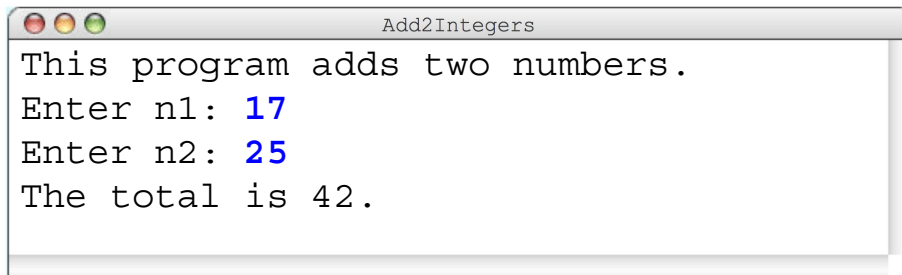
The screenshot shows a window titled "Add2Integers" with a white background and a grey title bar. The text inside the window is as follows:

```
This program adds two numbers.  
Enter n1: 17  
Enter n2: 25
```

Add2Integers

```
public class Add2Integers extends ConsoleProgram {  
    public void run() {  
        println("This program adds two numbers.");  
        int n1 = readInt("Enter n1: ");  
        int n2 = readInt("Enter n2: ");  
        int total = n1 + n2;  
        println("The total is " + total + ".");  
    }  
}
```

n1	n2	total
17	25	42

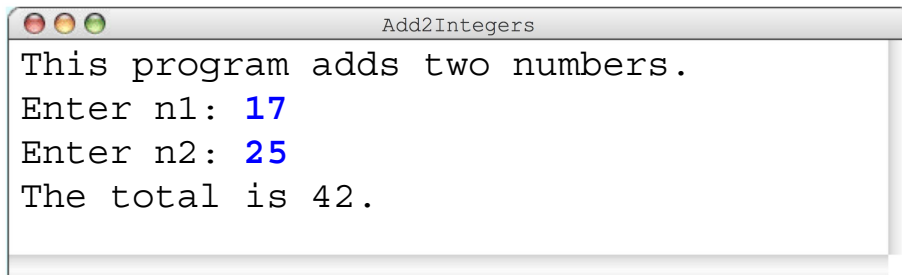


```
Add2Integers  
This program adds two numbers.  
Enter n1: 17  
Enter n2: 25  
The total is 42.
```


Add2Integers

```
public class Add2Integers extends ConsoleProgram {  
    public void run() {  
        println("This program adds two numbers.");  
        int n1 = readInt("Enter n1: ");  
        int n2 = readInt("Enter n2: ");  
        int total = n1 + n2;  
        println("The total is " + total + ".");  
    }  
}
```

n1	n2	total
17	25	42



The screenshot shows a window titled "Add2Integers" with the following text:

```
This program adds two numbers.  
Enter n1: 17  
Enter n2: 25  
The total is 42.
```

Compiler Errors

1. A variable can't be used until it is assigned a value

```
int x;  
println(x);           // Error: x has no value
```

2. You may not declare the same variable twice

```
int y = 3;  
int y = 5;         // Error: y already exists
```

3. You may not use a variable until it is declared

```
z = 10;             // Error: z cannot be resolved
```

Plan for Today

- Bye, Karel!
- Console programs
- Variables
- **Expressions**
- Practice: Receipt

Expressions

- You can combine literals or variables together into **expressions** using *binary operators*:

+	Addition	*	Multiplication
-	Subtraction	/	Division
		%	Remainder

Order of Operations

```
int result = 4 + 2 * 3;
```

Priority	Operator	Tie breaker
Highest	()	Left to right
Middle	* / %	Left to right
Lowest	+ -	Left to right

Order of Operations

```
// Multiplication before addition  
int result = 4 + 2 * 3;      // 10
```

Priority	Operator	Tie breaker
Highest	()	Left to right
Middle	* / %	Left to right
Lowest	+ -	Left to right

Order of Operations

// Multiplication before addition

```
int result = 4 + 2 * 3;    // 10
```

// Parens first, then left to right

```
int tot = 1 + 2 + (3 * 4); // 15
```

Priority	Operator	Tie breaker
Highest	()	Left to right
Middle	* / %	Left to right
Lowest	+ -	Left to right

Expressions

- You can combine literals or variables together into **expressions** using *binary operators*:

+ Addition

- Subtraction

* Multiplication

/ Division

% Remainder

What's THAT? %%%



%: Integer Remainder or Modulus

- The % operator computes the remainder from integer division

14 % 4 is 2

$$\begin{array}{r} \overline{) 14} \\ \underline{12} \\ \mathbf{2} \end{array}$$

218 % 5 is 3

$$\begin{array}{r} \overline{) 218} \\ \underline{20} \\ 18 \\ \underline{15} \\ \mathbf{3} \end{array}$$

%: Integer Remainder or Modulus

- The % operator computes the remainder from integer division

14 % 4 is 2

$$\begin{array}{r} 3 \\ \hline 4 \) \ 14 \\ \underline{12} \\ \textcircled{2} \end{array}$$

218 % 5 is 3

$$\begin{array}{r} 43 \\ \hline 5 \) \ 218 \\ \underline{20} \\ 18 \\ \underline{15} \\ \textcircled{3} \end{array}$$

- Applications of % operator:

- Obtain last digit of a number: 857 % 10 is 7
- Obtain last 3 digits: 26489 % 1000 is 489
- See if a number is odd or even: 7 % 2 is 1, but 42 % 2 is 0

What do you think this does?

```
double successRate = 1 / 2;
```

AHHHHH!

```
double successRate = 1 / 2;
```

0.0

successRate

Integer Division

- When we divide integers, the quotient is also an integer
 - $14 / 4$ is 3, not 3.5 (Java always rounds down)

Integer Division

- When we divide integers, the quotient is also an integer
 - 14 / 4 is 3, not 3.5 (Java always rounds down)

```
double successRate = 1 / 2;
```

Integer Division

- When we divide integers, the quotient is also an integer
 - 14 / 4 is 3, not 3.5 (Java always rounds down)

```
double successRate = 1 / 2;
```

The diagram illustrates integer division in a Java code snippet. The variable `successRate` is declared as a `double`. The expression `1 / 2` is evaluated as integer division, resulting in `0`. Red arrows point from the labels `int` below to the integers `1` and `2` in the expression, indicating that both operands are integers.

Integer Division

- When we divide integers, the quotient is also an integer
 - 14 / 4 is 3, not 3.5 (Java always rounds down)

```
double successRate = 1 / 2;
```

The diagram illustrates integer division in a Java code snippet. The expression `1 / 2` is enclosed in a red box. A red `0` is positioned above the box. Red arrows point from the `int` labels below to the `1` and `2` in the expression.

Integer Division

- When we divide integers, the quotient is also an integer
 - 14 / 4 is 3, not 3.5 (Java always rounds down)

`double successRate = 1 / 2;`

0

int int

1 / 2 → would be 0.5 → truncated to **0**

14 / 4 → would be 3.5 → truncated to **3**

199 / 100 → would be 1.99 → truncated to **1**

Type Interactions

`int` and `int` returns an `int` $7 / 2 \rightarrow 3$

`int` and `double` returns a `double` $7 / 2.0 \rightarrow 3.5$

`double` and `double` returns a `double` $4.4 * 0.5 \rightarrow 2.2$

* operations return the most expressive type

`String` > `double` > `int` > `char` > `boolean`

Pitfalls of Integer Division

Convert 100° Celsius temperature to its Fahrenheit equivalent:

```
double c = 100;
```

```
double f = 9 / 5 * c + 32;
```

Pitfalls of Integer Division

Convert 100° Celsius temperature to its Fahrenheit equivalent:

```
double c = 100;
```

```
double f = 9 / 5 * c + 32;
```



Pitfalls of Integer Division

Convert 100° Celsius temperature to its Fahrenheit equivalent:

```
double c = 100;
```

```
double f =  $\frac{9}{5}$  * c + 32;
```



Pitfalls of Integer Division

Convert 100° Celsius temperature to its Fahrenheit equivalent:

```
double c = 100;
```

```
double f = 9 / 5 * c + 32; // 132
```

1



Pitfalls of Integer Division

Convert 100° Celsius temperature to its Fahrenheit equivalent:

```
double c = 100;  
double f = 9 / 5 * c + 32; // 132
```

1



How can we fix it?

```
double c = 100;  
double f = 9.0 / 5 * c + 32;
```

Pitfalls of Integer Division

Convert 100° Celsius temperature to its Fahrenheit equivalent:

```
double c = 100;  
double f = 9 / 5 * c + 32; // 132
```

1



How can we fix it?

```
double c = 100;  
double f = 9.0 / 5 * c + 32; // 212.0
```

1.8

Pitfalls of Integer Division

Convert 100° Celsius temperature to its Fahrenheit equivalent:

```
double c = 100;  
double f = 9 / 5 * c + 32; // 132
```

1



How can we fix it?

```
double c = 100;  
double f = 9.0 / 5 * c + 32; // 212.0
```

1.8

* You can fix this problem by converting the fraction to a **double**, either by inserting decimal points or by using a type cast

Practice

- $5 + 3 / 2 - 4$
- $15 / 2.0 + 6$
- $1 * 2 + 3 * 5 \% 4$
- `"abc" + 1 + 2`
- `"abc" + (1 + 2)`

Practice

- $5 + 3 / 2 - 4$ 2
- $15 / 2.0 + 6$
- $1 * 2 + 3 * 5 \% 4$
- `"abc" + 1 + 2`
- `"abc" + (1 + 2)`

Practice

- $5 + 3 / 2 - 4$ 2
- $15 / 2.0 + 6$ 13.5
- $1 * 2 + 3 * 5 \% 4$
- "abc" + 1 + 2
- "abc" + (1 + 2)

Practice

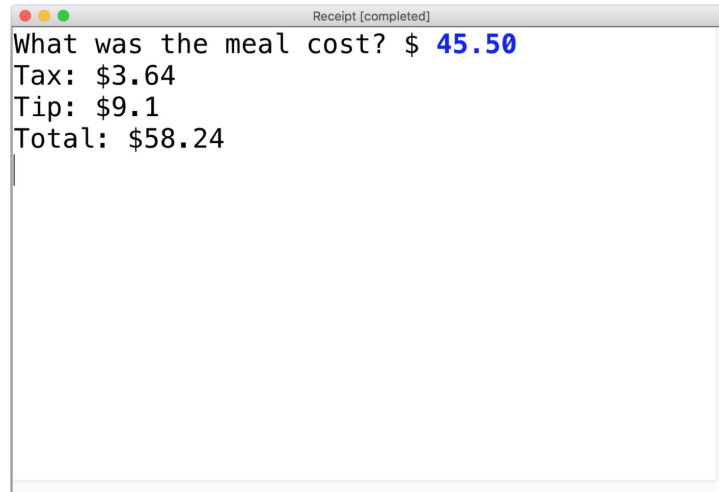
- $5 + 3 / 2 - 4$ 2
- $15 / 2.0 + 6$ 13.5
- $1 * 2 + 3 * 5 \% 4$ 5
- "abc" + 1 + 2 "abc12"
- "abc" + (1 + 2) "abc3"

Plan for Today

- Bye, Karel!
- Console programs
- Variables
- Expressions
- Practice: Receipt

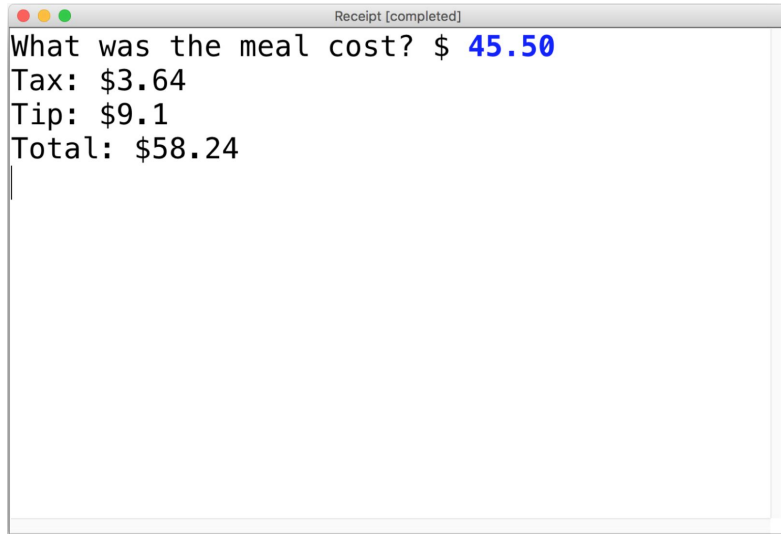
Practice: Receipt Program

- Let's write a ConsoleProgram that calculates the tax, tip and total bill for us at a restaurant.
- The program should ask the user for the subtotal, and then calculate and print out the tax, tip and total.



```
Receipt [completed]
What was the meal cost? $ 45.50
Tax: $3.64
Tip: $9.1
Total: $58.24
```

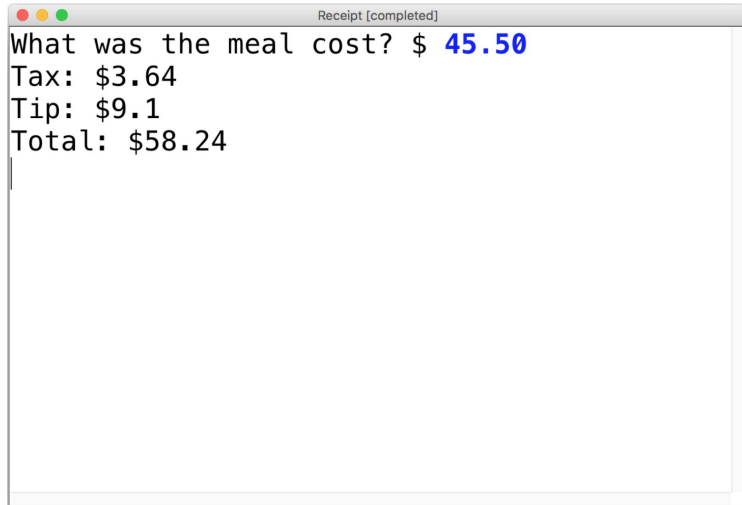
What's the pseudocode?



```
Receipt [completed]
What was the meal cost? $ 45.50
Tax: $3.64
Tip: $9.1
Total: $58.24
```

A terminal window titled "Receipt [completed]" displays the following text: "What was the meal cost? \$ 45.50", "Tax: \$3.64", "Tip: \$9.1", and "Total: \$58.24". The value "45.50" is highlighted in blue.

What's the pseudocode?



```
Receipt [completed]
What was the meal cost? $ 45.50
Tax: $3.64
Tip: $9.1
Total: $58.24
```

What's the Pseudocode?

Ask user for meal cost (store in variable)

Calculate tax (make variable!)

Calculate tip (make variable)

Calculate total (make variable)

Print out tax, tip, and total

Let's Code It!

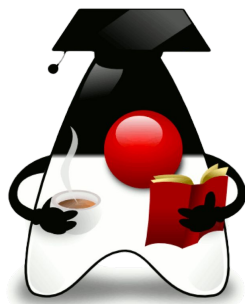
Practice: Receipt Program

```
public class Receipt extends ConsoleProgram {
    public void run() {
        double subtotal = readDouble("Meal cost? $");
        double tax = subtotal * 0.08;
        double tip = subtotal * 0.20;
        double total = subtotal + tax + tip;

        println("Tax: $" + tax);
        println("Tip: $" + tip);
        println("Total: $" + total);
    }
}
```

Recap

- Bye, Karel!
- Console programs
- Variables
- Expressions
- Practice: Receipt



Next time: Control flow in Java

[Extra] More on Precedence

- precedence: Order in which operators are evaluated.
 - Generally operators evaluate left-to-right.
 $1 - 2 - 3$ is $(1 - 2) - 3$ which is -4
 - But $*$ / $\%$ have a higher level of precedence than $+$ -
 $1 + 3 * 4$ is 13
 $6 + 8 / 2 * 3$
 $6 + 4 * 3$
 $6 + 12$ is 18
 - Parentheses can alter order of evaluation, but spacing does not:
 $(1 + 3) * 4$ is 16
 $1+3 * 4-2$ is 11

[Extra] String Concatenation

- string concatenation: Using + between a string and another value to make a longer string.

```
"hello" + 42    is "hello42"
```

```
1 + "abc" + 2   is "1abc2"
```

```
"abc" + 1 + 2   is "abc12"
```

```
1 + 2 + "abc"   is "3abc"
```

```
"abc" + 9 * 3   is "abc27"
```

```
"1" + 1         is "11"
```

```
4 - 1 + "abc"   is "3abc"
```

- Use + to print a string and an expression's value together.

```
println("Average: " + (95.1 + 71.9) / 2);    // Output: Average: 83.5
```

[Extra] Practice

- $5 + 3 / 2 - 4$
- $15 / 2.0 + 6$
- $1 * 2 + 3 * 5 \% 4$
- `"abc" + 1 + 2`
- `"abc" + (1 + 2)`

[Extra] Practice

- $5 + 3 / 2 - 4$ 2
- $15 / 2.0 + 6$ 13.5
- $1 * 2 + 3 * 5 \% 4$ 5
- "abc" + 1 + 2 "abc12"
- "abc" + (1 + 2) "abc3"

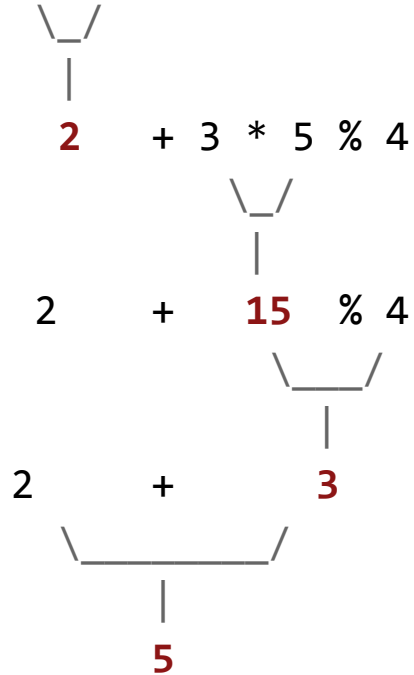
[Extra] Precedence Examples

1 * 2 + 3 * 5 % 4

“abc” + 1 + 2

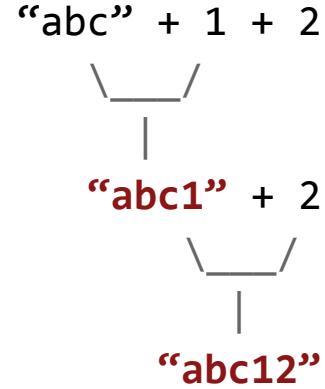
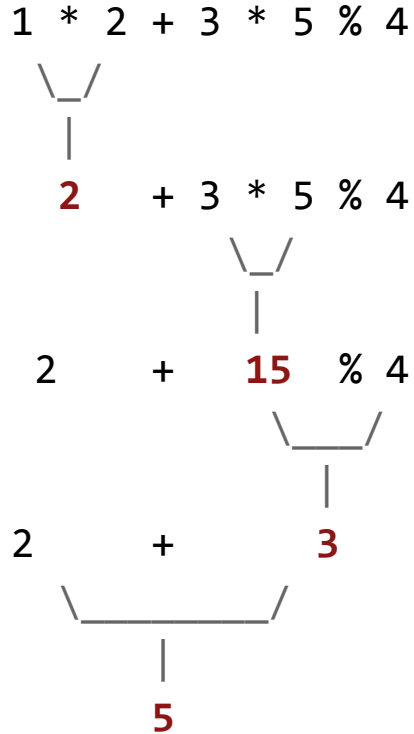
[Extra] Precedence Examples

1 * 2 + 3 * 5 % 4



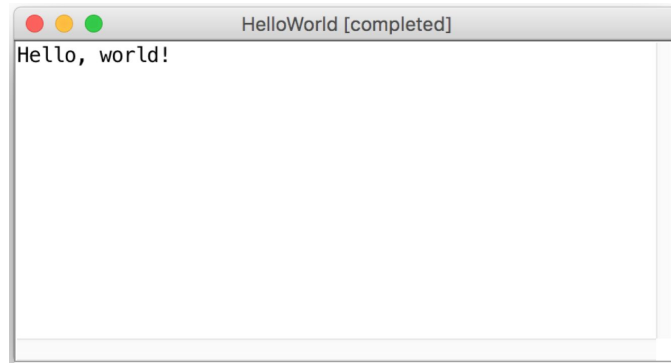
“abc” + 1 + 2

[Extra] Precedence Examples



[Extra] Print

```
public class HelloWorld extends ConsoleProgram {  
    public void run() {  
        print("Hello, ");  
        print("world!");  
    }  
}
```



Same as println, but does not go to the next line.

[Extra] Escape Sequences

- **escape sequence:** A special sequence of characters used to represent certain special characters in a string.

<code>\t</code>	tab character
<code>\n</code>	new line character
<code>\"</code>	quotation mark character
<code>\\</code>	backslash character

- Example:

```
println("\\hello\nhow\tare \"you\"?\\");
```

- Output:

```
\hello
how      are "you"?\\
```