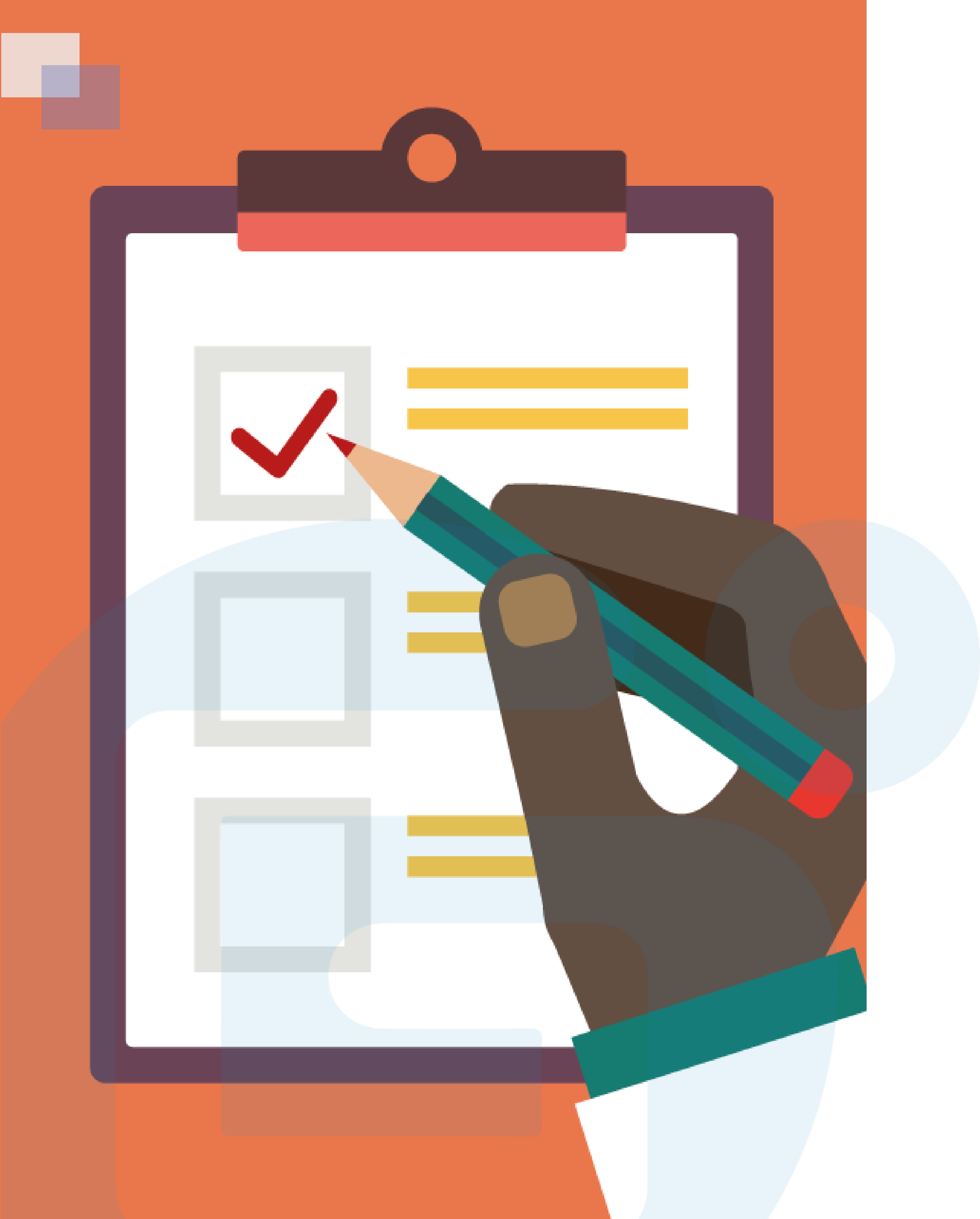
Lecture 2: Java Language Foundations I

Instructor: Xin Wang



CONTENTS OF THIS LESSON

- ➢ Syntax
- ➢ Output
- ➢ Comments
- ➢ Variables
- ➢ Operators
- ➢ Strings
- ➢ Math

Java Syntax

- In Java, every line of code muse be inside class **Main**.
- The main( ) method is required and any code inside the main( ) method will be executed.
- Every Java program has a class name that must match the filename. Every program must contain the main( ) method.

```
public static void main(String[] args)
```

System.out.println( )

Inside the main() method, the println() method can be used to print a line of text, as presented as below:

```
public static void main(String[] args) {
  System.out.println("Hello World");
}
```

Note:

-The beginning and the end of a block of code should have the curly braces { }

-System is a built-in Java class that contains useful members, such as output.

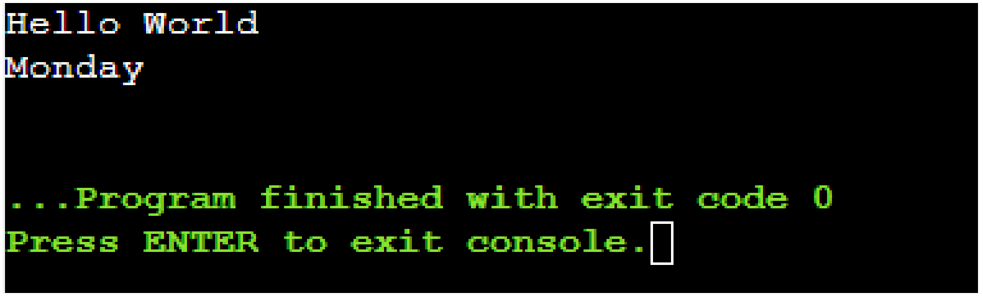
-println( )method represents “print line” and is used to print a result.

-Each code statement must end with a semicolon.

Java Output

You can print out more results, using println( ) method.

```
public static void main(String[] args) {
  System.out.println("Hello World");
  System.out.println("Monday");
}
```



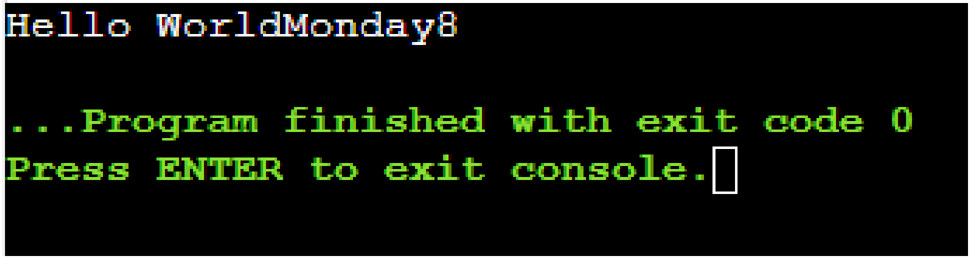
```
Hello World
Monday


...Program finished with exit code 0
Press ENTER to exit console.
```

You also can do mathematical calculations.

```
System.out.println(3 + 3);
```

print( )method will not add a line.

```
public static void main(String[] args) {
  System.out.print("Hello World");
  System.out.print("Monday");
  System.out.print(1+7);
}
```



```
Hello WorldMonday8

...Program finished with exit code 0
Press ENTER to exit console.
```

Java Comments

Single-line Comments

- Single-line comments start with two forward slashes //.
- Single-line comment can be used before a code line.
- Single-line comment can be used at the end of a code line.

```
// This is a comment
System.out.println("Hello World"); // This is a comment
```

Multi-line Comments

- Multi-line comments start with /* and ends with */.

```
/* The code below will print the words Hello Worldto the screen, and it is amazing */
System.out.println("Hello World");
```

Java Variables

Variables: containers for storing data values

Types of variables:

String – stores text. String values are surrounded by double quotes. Eg. “Hello”.

int – stores integers. Eg. 87

float – stores floating point numbers. Eg. 453.89

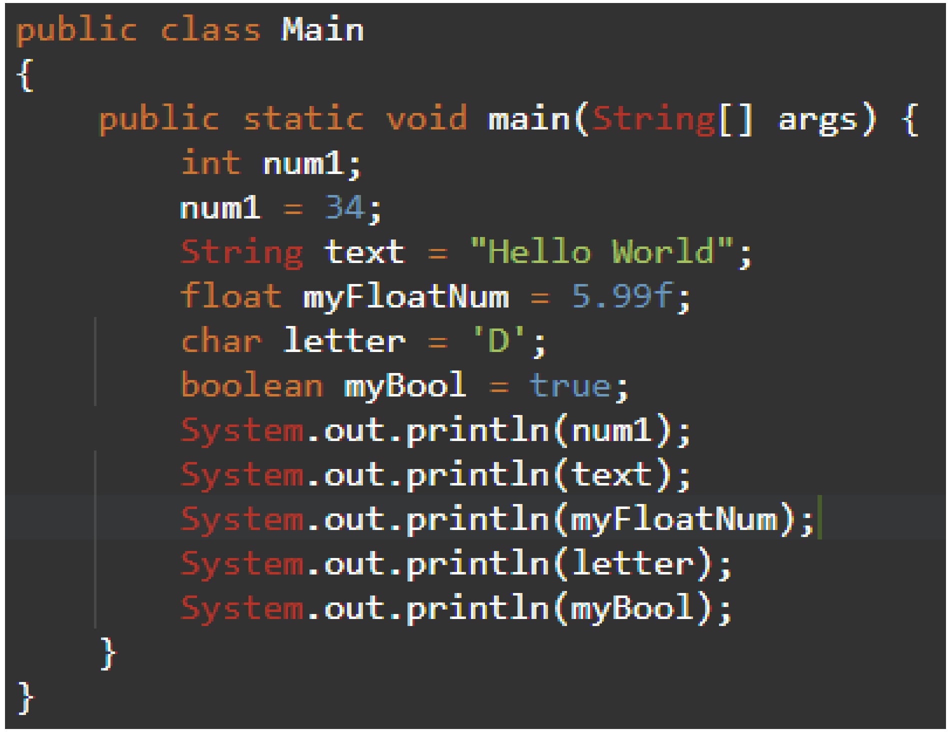
char – stores single characters. Eg. ‘e’. (Note: Char values are surrounded by single quotes)

boolean – stores values with two states: true or false

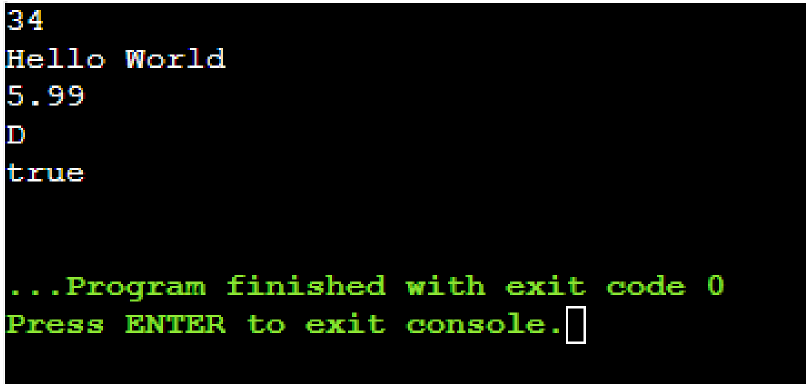
Declaring Variables: Specify the variable type and assign a value for it.

Syntax:

```
type variableName = value;
```



```
public class Main
{
    public static void main(String[] args) {
        int num1;
        num1 = 34;
        String text = "Hello World";
        float myFloatNum = 5.99f;
        char letter = 'D';
        boolean myBool = true;
        System.out.println(num1);
        System.out.println(text);
        System.out.println(myFloatNum);
        System.out.println(letter);
        System.out.println(myBool);
    }
}
```



```
34
Hello World
5.99
D
true


...Program finished with exit code 0
Press ENTER to exit console.
```

Q: What will be the result?

```
int myNum = 15;
myNum = 20;  // myNum is now 20
System.out.println(myNum);

int x = 5;
int y = 6;
System.out.println(x + y); // Print the value of x + y

String name = "John";
System.out.println("Hello " + name);

String firstName = "John ";String lastName = "Doe";
String fullName = firstName + lastName;
System.out.println(fullName);
```

Tips:

- For numeric values, + character works as a mathematical operator.
- + character can be used to combine both text and a variable.
- + character can add a variable to another variable.

The rules for variable name:

- Names must begin with a lowercase letter or characters, such as $ and _.
- Names are case sensitive. (Eg. “myNum” and “mynum” are different variables).
- Reserved words (Java keywords), such as int or boolean, cannot be used as name.

Java Operators

Types of operators in Java:

- Arithmetic operators - used to perform common mathematical operations.
- Assignment operators – used to assign values to variables.
- Comparison operators – used to compare two values.
- Logical operators – used to determine the logic between variables or values.
- Bitwise operators – used to perform the manipulation of individual bits of a number.

Arithmetic operators

Operator	Name	Description	Example
+	Addition	Adds together two values	x + y
-	Subtraction	Subtracts one value from another	x - y
*	Multiplication	Multiplies two values	x * y
/	Division	Divides one value by another	x / y
%	Modulus	Returns the division remainder	x % y
++	Increment	Increases the value of a variable by 1	++x
--	Decrement	Decreases the value of a variable by 1	--x

Assignment Operators

Operator	Example	Same As
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3

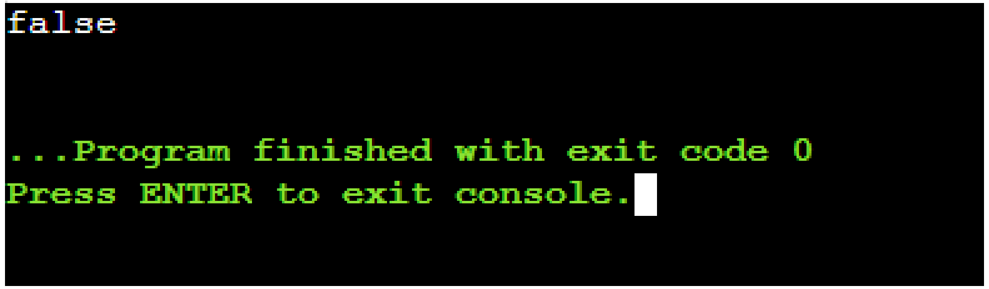
```java
int x = 10;
x %= 5;
System.out.println(x);
```

```java
int x = 10;
x /= 5;
System.out.println(x);
```

Comparison Operators

Operator	Name	Example
==	Equal to	x == y
!=	Not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

```java
public static void main(String[] args) {
  int x = 10;
  float y = -35.55f;
  System.out.print(x<y);
}
```



```
false


...Program finished with exit code 0
Press ENTER to exit console.
```

Logical Operators

Operator	Name	Description	Example
&&	Logical and	Returns true if both statements are true	x < 5 && x < 10
||	Logical or	Returns true if one of the statements is true	x < 5 || x < 4
!	Logical not	Reverse the result, returns false if the result is true	!(x < 5 && x < 10)

```
int x = 8;
System.out.println(x<5&&x<10);
```

```
int x = 8;
System.out.println(!(x<5&&x<10))
;
```

```
double x = 4.5;
System.out.println(x<5||x<4);
```

Java Strings

String methods

Method	Function
length()	To find the length of a string
toUpperCase()	To convert all the characters in a string to uppercase
toLowerCase()	To convert all the characters in a string to lowercase
indexOf()	To return the index of the first occurrence of a specified text in a string

```java
String txt = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
System.out.println("The length of the txt string is: " + txt.length());

String txt = "Hello World";
System.out.println(txt.toUpperCase());
System.out.println(txt.toLowerCase());

String txt = "Please locate where 'locate' occurs!";
System.out.println(txt.indexOf("locate"));
```

String Concatenation

concat( ) method can be used to concatenate two strings

```
String firstName = "John ";String lastName = "Doe";
System.out.println(firstName.concat(lastName));
```

Adding number and string

```
String x = "10";String y = "20";
String z = x + y;  // z will be 1020 (a String)
```

```
String x = "10";int y = 20;
String z = x + y;  // z will be 1020 (a String)
```

Special Characters

Escape character	Result	Description
\'	'	Single quote
\"	"	Double quote
\\	\	Backslash

```
String txt = "We are the so-called \"Wild Wine\" from the north. It\'s alright.\\";
```

Java Math

Java Math Class

Math Class	Function
Math.max(*x,y*)	To find the highest value between x and y.
Math.min(x,y)	To find the lowest value between x and y.
Math.sqrt(x)	To calculate the square root of x.
Math.abs(x)	To get the absolute value of x.
Math.random( )	To return a random number between 0.0 (inclusive) and 1.0 exclusive)

```java
int randomNum = (int)(Math.random() * 101);  // 0 to 100
```

Q: How to find the highest value/ lowest value among multiple values using Java?

Thank you!
Any questions?