

CS1101

Programming and

Problem Solving

Dr. Gina Bai
Spring 2023



Logistics

- **HW0-A, HW0-B** on Brightspace > First Week
 - Due: Today, at 11:59 pm
- **PA00-A** and **PA00-B** on zyBook > Chap 11
 - Due: Thursday, Jan 19, at 11:59pm
- **ZY-1** and **ZY-2A** on zyBook > Assignments
 - Due: Saturday, Jan 21, at 11:59pm

MLK Day

- Monday, Jan 16
- NO CLASS

Recap – HelloWorld.java

- Class – A program
 - Method – A named group of statements
 - Statement – An instruction to be executed

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

Recap – Basics of Java

Q: Identifiers cannot start with a _____?

- A. Uppercase letter
- B. Lowercase letter
- C. Underscore symbol
-  D. Digit
- E. None of above

Recap – Basics of Java

1. A(n) **program** is a collection of statements written in a programming language.
2. A(n) **algorithm** is a step-by-step list of instructions for solving a problem and is independent of any programming language.
3. A(n) **compiler** translates Java source code to bytecode.
4. A(n) **main** method is required for an executable Java program.
5. A line that begins with // is a(n) **comment**
6. True/False: A .class file is executable with JVM. **True**

Print Statements

`System.out.println("TEXT");`

- Print “TEXT”, and the cursor **moves to the next line** at the console

`System.out.print("TEXT");`

- Print “TEXT”, the cursor **remains at the end** of that text in the console

Q: What does `System.out.println()` print?
A blank line of output

Q: What's the exact output for this?

```
public class PrintDemo {  
    public static void main(String[] args) {  
        System.out.print("Hello");  
        System.out.println("!");  
        System.out.println();  
        System.out.println("This is");  
        System.out.print(" CS1101" + ".");  
    }  
}
```

**String Concatenation
with + symbol**

```
$ javac PrintDemo.java  
$ java PrintDemo  
Hello!
```

This is
CS1101.



Debugging

zyBook Chap 1.7, 2.18

Debugging & Strategies

- Debug code → Examine the code to figure out what went wrong, and **remove errors** from the code
- Do not make random or extensive changes to the program!
- Think before you change anything
- Figure out *why* it went wrong

Reacting to Errors

- Always address the **first error listed in the compiler output** before addressing the other errors.
- Always **recompile after making changes to code** because fixing one error will often fix other errors as well.



The screenshot shows a Mac OS X desktop environment. At the top, there is a window titled "HelloWorld.java" containing Java code. Below it is a terminal window titled "Desktop -- zsh -- 84x12". The terminal displays the output of the javac compiler, showing three errors in the HelloWorld.java file. The errors are:

```
ginabai@MacBook-Air-104 Desktop % javac HelloWorld.java
HelloWorld.java:2: error: ';' expected
    public static void main(String[] args) {
                           ^
HelloWorld.java:3: error: class, interface, enum, or record expected
        System.out.println("Hello World!");
                           ^
HelloWorld.java:4: error: class, interface, enum, or record expected
    }
                           ^
3 errors
ginabai@MacBook-Air-104 Desktop %
```

A yellow callout box labeled "Compiler Output" points to the terminal window.



HelloWorld > No Selection

```
1 public class HelloWorld {  
2     public static void main(String[] args) }  
3     System.out.println("Hello World!");  
4 }  
5 }
```

File name and line number

```
[ginabai@MacBook-Air-104 Desktop % javac HelloWorld.java  
HelloWorld.java:2: error: ';' expected  
    public static void main(String[] args) }  
                                ^  
  
HelloWorld.java:3: error: class, interface, enum, or record expected  
    System.out.println("Hello World!");  
                           ^  
  
HelloWorld.java:4: error: class, interface, enum, or record expected  
    }  
    ^  
3 errors  
ginabai@MacBook-Air-104 Desktop %
```

A red arrow points from the text "File name and line number" to the line "HelloWorld.java:2: error: ';' expected". Another red arrow points from the text "Suggested error" to the character '^' in the code editor.

Program Errors

zyBook 1.7

Three Types of Program Errors

- Syntax Errors / Compilation Errors
 - The code does not compile
- Runtime Errors
 - The code compiles, does not run
- Logic Errors
 - The compiles and runs, but the output is not correct

Syntax Errors / Compilation Errors

- A grammatical mistake in a program
- **Caught by the compiler**
- Examples:
 - Missing semicolon
 - Missing or mismatching curly braces
 - Missing keyword
 - Misspelling a keyword or identifier
 - Incorrect class name (mismatches the file name)

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello World!")  
    }  
}  
  
$ javac HelloWorld.java  
HelloWorld.java:3: error: ';' expected  
        System.out.println("Hello World!")  
                           ^  
1 error
```

Runtime Errors

- The program is grammatically correct and hence can be compiled, but the program **cannot be run**
- Examples
 - Missing main method
 - Incorrect spelling for main
 - Dividing by 0

```
public class HelloWorld {  
    public static void Main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}  
  
$ javac HelloWorld.java  
$ java HelloWorld  
Error: Main method not found in class HelloWorld,  
please define the main method as:  
    public static void main(String[] args)  
or a JavaFX application class must extend  
javafx.application.Application
```

Logic Errors

- The program **compiles and runs** but it does not perform the task it is intended to perform
- Strategy → **Test programs thoroughly!**

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("2 + 3 = " + 2 * 3);  
    }  
}
```

```
$ javac HelloWorld.java  
$ java HelloWorld  
2 + 3 = 6
```

Data Types

zyBooks Chap 2.5, 2.6, 2.7, 2.8

Basics

What is a data type?

- A name for a category of data values are related

Why do we care about data types?

- Java is “strongly typed”, which means that each piece of data that we use has a specific type

Data Types in Java

Java supports **two types** of data

- **Primitive data (8 types)**
- Objects, consist of fields and methods

Primitive Data Types

Type	Supported Number Range
byte	whole numbers from -128 to 127
short	whole numbers from -32,768 to 32,767
int	whole numbers from -2,147,483,648 to 2,147,483,647
long	whole numbers from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
float	Sufficient for storing 6 to 7 decimal digits
double	(default) Sufficient for storing 15 decimal digits
char	A single character/letter or ASCII values
boolean	true / false

Q: What primitive data type would you use to store...

1. whether a restaurant is open?
2. a person's middle initial?
3. the number of students in class?
4. cost of a meal?
5. distance to campus?
6. the number of siblings a person has?
7. whether you pass a course?
8. your grade in a class?