

CS1101

Programming and

Problem Solving

Dr. Gina Bai
Spring 2023



Logistics

- **PA00 - A & B** are posted on zyBook > Chap 11

- Tutorials on zyLabs
- Due: Thursday, **Jan 19**, at 11:59pm

The screenshot shows a section titled "11. Programming Assignments (PAs) | zyLabs". It contains two items: "11.1 PA00-A: Getting to Know zyLabs: The Basics" and "11.2 PA00-B: Getting to Know zyLabs: Dealing with Input/Output". Each item has a green "Lab" button next to it.

11.1 PA00-A: Getting to Know zyLabs: The Basics	Lab
11.2 PA00-B: Getting to Know zyLabs: Dealing with Input/Output	Lab

- **ZY-1** and **ZY-2A** are posted on zyBook > Assignments

- Due: Saturday, **Jan 21**, at 11:59pm

The screenshot shows an "Assignments" page with two assignments listed: "ZY-1" and "ZY-2A". Both assignments are due on "01/21/2023, 11:59 PM CST". The points for each assignment are 120 pts and 98 pts respectively. There are "Add" and "Report" buttons above the list.

Assignments	+ Add	Report
ZY-1 Due: 01/21/2023, 11:59 PM CST	120 pts	
ZY-2A Due: 01/21/2023, 11:59 PM CST	98 pts	

Logistics

- **HW 0–A & 0–B** on Brightspace > First Week
 - Due: Friday, **Jan 13**, at 11:59 pm

Visit  tophat.com/students/

Join Code

Section 1 (10:10am): **499349**

Section 2 (11:15am): **144507**

Programming (General)

zyBooks Chap 1.3

Terminologies

- Computer Science
 - The study of **algorithms** for processing information
- Algorithms
 - A **precise, unambiguous, step-by-step** method for completing a task in a **finite** amount of time

Terminologies

- Program
 - A set of instructions for a computer to follow
 - Get the input/data → Process the input → Produce the output
- Programming Language
 - A set of **rules** which allow programmers to describe an algorithm in a way that a computer can understand

Structure of Java

zyBooks Chap 1.5, 1.6, 2.1, 2.2

Structure of Java

- Class
 - Method
 - Statement
- Documentation/Comments
- Spacing

```
/**  
 * This is an example class illustrates printing a message to the screen.  
 *  
 * @author Gina Bai  
 */  
public class HelloWorld {  
  
    // This is the main method  
    // A "MUST-HAVE" method in every executable program  
public static void main(String[] args) {  
  
    // This is a print statement  
    System.out.println("Hello World!");  
}  
}
```

Documentation/Comments

- A **note** for programmers that **describes** or **clarifies** the code
 - **Not readable** to computers
 - **NOT executed** when the program runs

Javadoc → `/**.....*/`

```
/*
 * This is an example class illustrates printing a message to the screen.
 *
 * @author Gina Bai
 */
public class HelloWorld {

    // This is the main method
    // A "MUST-HAVE" method in every executable program
    public static void main(String[] args) {

        // This is a print statement
        System.out.println("Hello World!");
    }
}
```

Single-line Comments → `//`

Read more in Programming Style Guide (Brightspace | Content | Course Documents)

Class – A program

- Class Header
 - Capitalize the first letter of each word, no space in between (e.g. HelloWorld)
- The **file name** (HelloWorld.java) must **match exactly** with the **class name**, including capitalization
 - Java is **case-sensitive!!!**

```
/**  
 * This is an example class illustrates printing a message to the screen.  
 *  
 * @author Gina Bai  
 */  
public class HelloWorld {  
  
    // This is the main method  
    // A "MUST-HAVE" method in every executable program  
public static void main(String[] args) {  
  
    // This is a print statement  
    System.out.println("Hello World!");  
}
```

Matching braces{ ... }

Method – A named group of statements

- Method Header
 - Begin with a lowercase letter, capitalize the first letter of the attached words
- Every **executable** Java program consists of a class, that **contains** a method named **main** that contains the statements to be executed

```
/**  
 * This is an example class illustrates printing a message to the screen.  
 *  
 * @author Gina Bai  
 */  
public class HelloWorld {  
  
    // This is the main method  
    // A "MUST-HAVE" method in every executable program  
public static void main(String[] args) {  
  
    // This is a print statement  
    System.out.println("Hello World!");  
}  
}
```

Matching braces { ... }

Statement – An instruction to be executed

- **Ends with semi-colon (;)**

```
/**  
 * This is an example class illustrates printing a message to the screen.  
 *  
 * @author Gina Bai  
 */  
public class HelloWorld {  
  
    // This is the main method  
    // A "MUST-HAVE" method in every executable program  
public static void main(String[] args) {  
  
    // This is a print statement  
    System.out.println("Hello World!");  
}  
}
```

Print/Println Statement

- `System.out` is an **object** for sending output to the screen
- `println` is a **method** to print whatever is **inside parentheses** to the screen, in this case, a String "Hello World!"
 - The item(s) inside parentheses are called **parameter(s)** or **argument(s)**

```
/**  
 * This is an example class illustrates printing a message to the screen.  
 *  
 * @author Gina Bai  
 */  
public class HelloWorld {  
  
    // This is the main method  
    // A "MUST-HAVE" method in every executable program  
public static void main(String[] args) {  
  
    // This is a print statement  
    System.out.println("Hello World!");  
}  
}
```

Spacing

- The best way to make your code readable is to **indent** nested code
- Indent every time you **go inside braces**
- You must indent using **four spaces** or **one tab** (manually set it to four spaces)

```
/**  
 * This is an example class illustrates printing a message to the screen.  
 *  
 * @author Gina Bai  
 */  
public class HelloWorld {  
  
    // This is the main method  
    // A "MUST-HAVE" method in every executable program  
    public static void main(String[] args) {  
  
        // This is a print statement  
        System.out.println("Hello World!");  
    }  
}
```

Read more in Programming Style Guide (Brightspace | Content | Course Documents)

Identifier

- Identifier is a **name** given to an entity in a program, such as a class name
- Identifiers **start with a letter** and are followed by a number of letters or digits. Letters include:
 - Alphabetic characters, upper and lower case (A-Z, a-z)
 - Underscore (_)
 - Dollar sign (\$)
- Identifiers should be descriptive/meaningful

Q: Indicate if each of the following Java identifier is legal or not.
If not, why?

- 3Example  Starts with a digit
- varTest 
- max_value 
- max-value  Contains -
- quiz+HW  Contains +
- quiz 1  Contains space
- system 

Java Keywords

An identifier that you cannot use because it already has a **reserved** meaning in Java.

abstract	default	if	private	this
boolean	do	implements	protected	throw
break	double	import	public	throws
byte	else	instanceof	return	transient
case	extends	int	short	try
catch	final	interface	static	void
char	finally	long	strictfp	volatile
class	float	native	super	while
const	for	new	switch	
continue	goto	package	synchronized	

Write-Compile-Execute

Write

Write

HelloWorld.java

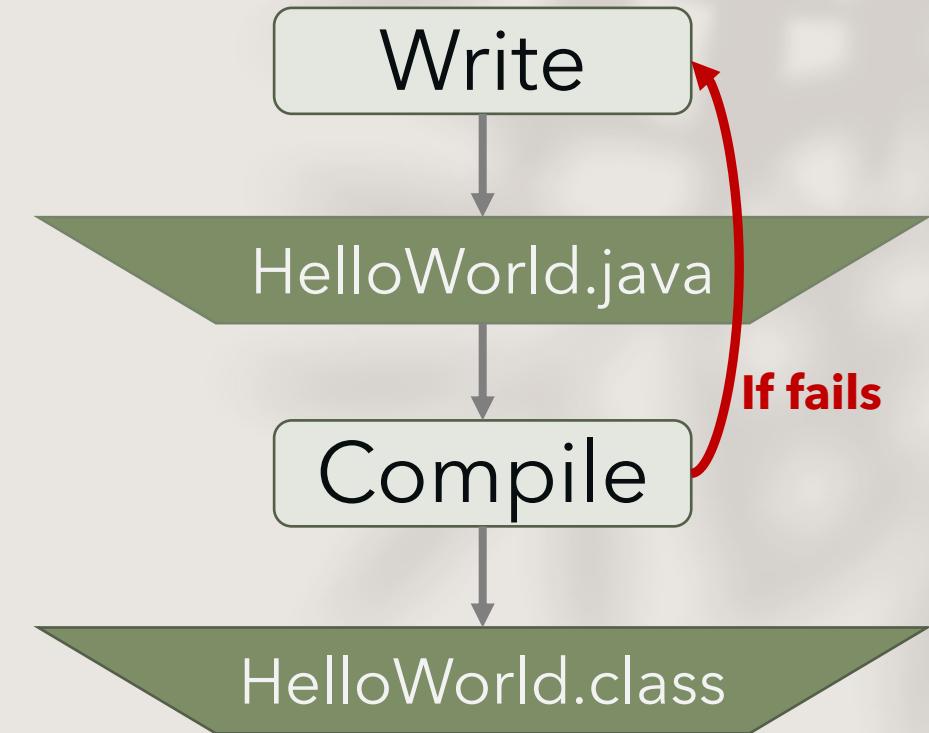
Write source code (instructions in a program)

- Written using a programming language (Java)
 - Human readable
- Written within a text editor/IDE
- The file extension will be **.java**

Compile

Translates program from one language to another

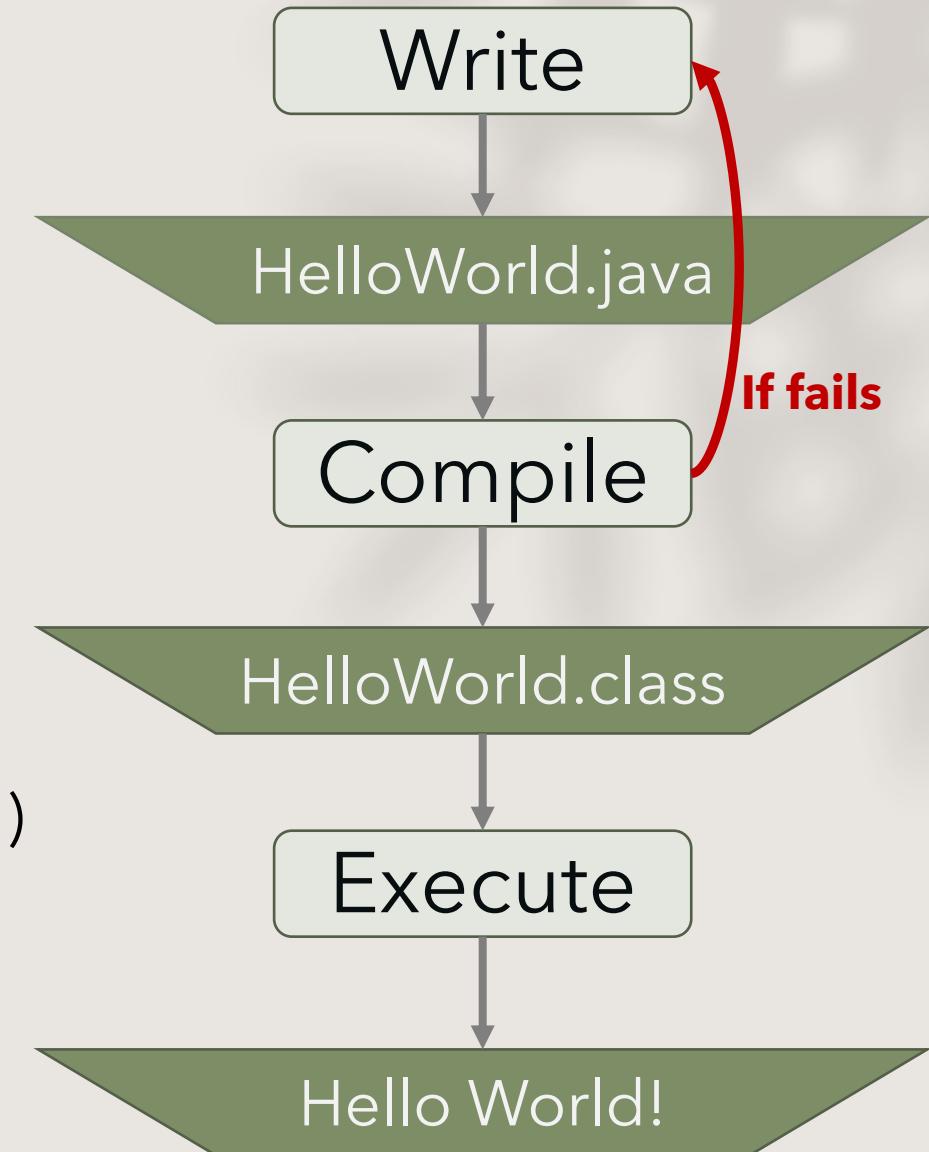
- High-level → high-level / low-level
- For Java, it translates the program to **byte-code** (computer object code), which is an **intermediate, low-level** language that can be run by many computers
- The file extension will be **.class**



Execute

Uses the **Java Virtual Machine (JVM)** to run/**interpret** the program

- **JRE**: Java Runtime Environment, a software package that provides Java class libraries, loader class, and JVM
 - **JVM**: A theoretical computer (specification) that provides a runtime environment in which java bytecode can be executed



Java Development Kit (JDK)

Contains

- Compiler
 - Translates Java to bytecode
- Interpreter – JVM
 - Translates bytecode to binary machine code (0s and 1s)
 - Carries out the instructions

Unix Commands in Terminal

- To compile

javac HelloWorld.java

where the HelloWorld.java is the file name

- To execute

java HelloWorld

where the HelloWorld is the class name

cs1101 – HelloWorld.java [lecExample]

Match exactly

Project

cs101 ~/IdeaProjects/cs101
.idea
lecExample
src
HelloWorld
HelloWorld.java
lecExample.iml
out
src
cs101.iml
External Libraries
Scratches and Consoles

>HelloWorld.java

```
1 public class HelloWorld {  
2     public static void main(String[] args) {  
3         System.out.println("Hello World!");  
4     }  
5 }  
6 }
```

Click here to run

Make sure it's the correct Java file

Make sure the java file is in the correct folder or file path

1) The cursor
2) When placing the cursor after a closing curly brace, its matching opening curly brace will be highlighted as well

Run: HelloWorld

/Library/Java/JavaVirtualMachines/jdk-9.0.4.jdk/Contents/Home/bin/java -javaagent:/Application
Hello World!

See output in the console/terminal

Indentation

The cursor is at line 4 position 6

All files are up-to-date (3 minutes ago)