GUIDED NOTES

# Your First Java Program

Name: _____ Date: _____

*Fill in the blanks as we go. Each day has its own word bank, and every answer is in it.*

**WHAT YOU ALREADY KNOW**

A program runs its steps in order, from top to bottom. A program that shows Hi and then Bye shows Hi on the first line and Bye on the second.

The text a program shows sits inside quote marks, and the quote marks never print. A program that shows "Welcome!" prints Welcome! with no quote marks.

A compile error message names a line. When the message says line 2, go to line 2 and look for the mistake there.

## Day 1 — Every Java Program Lives in a Class

*Today you will learn the parts of a Java program: a class, a `main` method where Java starts, and statements that print text with `System.out.println()` and `System.out.print()`.*

**Word bank** Braces · class · command line · comment · main method · semicolon · statement

### The parts of a Java program

*Every Java program lives in a class, and Java starts running it in `main`.*

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

**Output**

```
Hello!
```

Java code lives in a _____. `public class Main {` starts one named `Main`, saved in a file named `Main.java`.

Java starts running the program in the _____. Its first line is `public static void main(String[] args) {`.

_____ mark what is inside. Each `{` opens a part, and its `}` closes it.

### Words from the command line

`String[] args` *holds the words typed after the program's name.*

The _____ is where you type a command to run a program, like java Main.

The words typed after java Main are called arguments. `String[] args` holds them.

The command java Main gives `args` no words. The command java Main red 7 gives `args` 2 words: red and 7.

### Statements, semicolons and comments

*Each step is a statement, and Java skips comments.*

```
1  // Ahmed's to-do list
2  System.out.println("Feed the cat.");
3  /* System.out.println("Nap."); */
4  System.out.println("Read.");
```

**Output**

```
Feed the cat.
Read.
```

A _____ is one step in a program, like one print line.

A print line ends with a _____, the `;` mark. Java doesn't end a statement at a line break.

A _____ is a note for people, and Java skips it. `//` starts one that runs to the end of the line.

A block comment starts with `/*` and ends with `*/`. It can turn off a line of code, like line 3 above.

### Printing text

`System.out.println()` *and* `System.out.print()` *show text on the screen.*

```
1  System.out.print("Hello, ");
2  System.out.println("Beatrice!");
3  System.out.print("Welcome to class.");
```

### What prints?

**Output**

Text to print goes inside double quotes. The quotes don't print.

`System.out.println()` prints its text and then ends the line.

`System.out.print()` prints its text and stays on the same line. The next text prints right after it.

Write what this code prints in the box.

**REMEMBER**

Every Java program lives in a class, and Java starts it in **`main`**. A print line ends with **`;`**, and Java skips comments. **`println`** ends the line, and **`print`** stays on it.

# Day 2 — Java Compiles Before It Runs

*Today you will learn that Java compiles the whole file before it runs, and that a compile error names the line where you start looking.*

**Word bank** compile · compile error · compiler · error message · run

## Compile, then run

*Java checks the whole file before it runs any line.*

To _____ means to check the whole file and turn it into code the computer can run.

The _____ does this job. It reads every line before the program starts.

Java can _____ a program only after it compiles. Then the steps go from top to bottom.

When you compile Java, the compiler makes `.class` files full of bytecodes.

The Java Virtual Machine (JVM) runs the bytecodes. The JVM works on many operating systems, so the same `.class` files can run on many kinds of computers.

## Compile errors

*A compile error stops the program before it starts.*

```
1  System.out.println("Thank you!");
2  System.out.println("Come again.);
```

**Error**

**Line 2:** Java says: `error: unclosed string literal`

A _____ is a mistake that stops the compiler, like a missing `;` or a missing quote mark.

With just one of these mistakes, nothing runs, so nothing prints. Even the lines before the mistake don't print.

The _____ for a compile error names the file and a line. Start there, then check the line just above it.

Java cares about capital letters. `system.out` and `printLn` are compile errors.

**REMEMBER**

Java compiles the whole file before it runs any line. One compile error stops everything, and its message names the line where you start looking.

# Day 3 — Quiz: Java Program Parts and Compile Errors

*Today you will review the parts of a Java program, `print` and `println`, and compile errors, then show what you know on the quiz.*

**Word bank** class · comment · compile · compile error · main method

### Review: print and println

*`println` ends the line, and `print` stays on it.*

`System.out.print()` prints its text and stays on the same line.

`System.out.println()` prints its text and then starts a new line.

To print two pieces of text on one line, use `print` for the first piece.

### Review: compile errors and the parts of a program

*Java checks the whole file, then starts in `main`.*

```java
// A note for the class
public class Main {
    public static void main(String[] args) {
        System.out.println("Quiz day!");
    }
}
```

**Output**

```
Quiz day!
```

Java cares about capital letters. `system.out` with a small s is a _____.

Java can't read past a missing `;`, so it may show only that first error.

Fix it, then _____ again.

A _____ is a note for people. Java skips it.

The _____ holds the _____, and Java starts running there.

**REMEMBER**

**`print`** stays on the line, and **`println`** ends it. Java compiles the whole file first, so fix each compile error and compile again.