

PLTW

CSA

Activity 1.1.1 Getting Started with Java

PLTW

GETTING STARTED WITH JAVA

- Java Programming Language
- Compiler and Runtime Errors
- The main Method
- Printing and Escape Sequences
- It's Your Turn
- Conclusion

Java Programming Language

GOALS

RESOURCES

- Learn about the Java® programming language and create your first Java program.
- Learn some basic rules of Java programming by identifying and correcting errors in code.
- Generate outputs to a console by calling `System` class methods.
- Understand the difference between the `print()` method and the `println()` method.

GOALS

RESOURCES

- [Computer Science Notebook](#)
- [Interactive Code Editor](#)

- [**Activity 1.1.1 Getting Started with Java \(Downloadable PDF\)**](#)

Why Java? Java is a widely used, platform-independent programming language. Using a [**Java Virtual Machine \(JVM\)**](#), a Java program can run on various platforms, including websites, phones, and video game consoles. You can use Java to create many different types of programs, including mobile applications and computer applications like music-mixing software, money management systems, and games, to name a few.

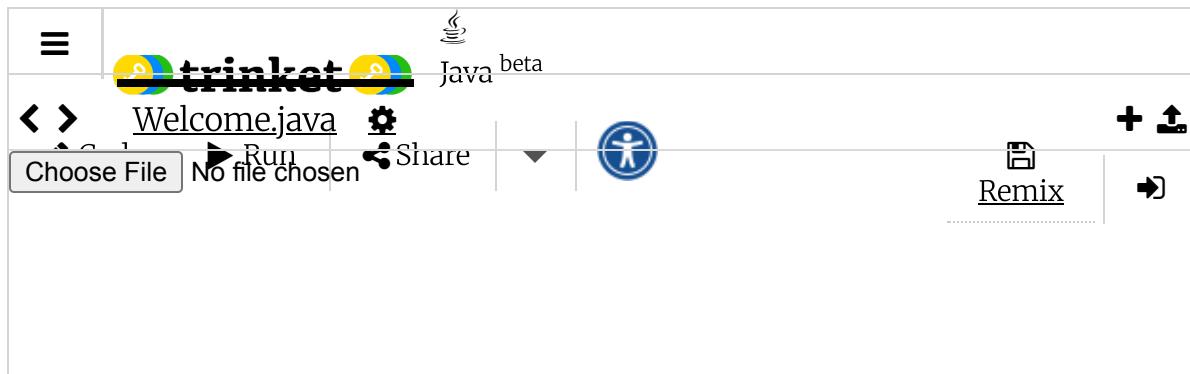
These first simple programs are intended to introduce a few key concepts related to the Java programming language. You will learn more about the parts of a Java program later in the lesson. For now, focus on uncovering some Java programming rules by looking at errors generated by incorrect code.

Following is an example that will introduce you to some fundamental parts of a Java program.

1

Examine the code below.

HELPFUL TIP: To view the code in a new tab, click the Trinket icon (the yellow key icon) and select **View on trinket.io**.



The lines beginning with `public` are considered meta instructions and define the structure of the program. The two lines inside the curly braces beginning with `System` are the sequence of steps the program actively completes. Each line is processed individually.

A **block of code** is any section of code enclosed in curly braces. An example of a block of code is the sequence of the two `System` statements in the `Welcome` program.



Interactive Code Editor

- Sometimes you will be asked to make changes in the code editor. If you ever make a change and cannot fix it, you can reset the code in the editor by opening the hamburger menu button on the top left and selecting **Reset**.

The screenshot shows the trinket Java beta interface. On the left, a sidebar titled 'MENU' contains options: 'Reset' (highlighted with a red box and a large orange arrow pointing to it), 'Fullscreen', 'Download', and 'Settings'. The main area is titled 'Welcome.java' and contains the following Java code:

```
1 public class Welcome
2 {
3     public static void main(String[] args)
4     {
5         System.out.println("Welcome to PLTW Computer Science A.");
6         System.out.println("Hope you have fun learning Java!");
7     }
8 }
9
```

- You cannot save your work directly in the interactive code editor. To keep a copy of your work, you must download your program or copy the code to a file on your computer.
- For more information, refer to the [Interactive Code Editor](#) resource.

2

Click on the **Run** button to run your program and view the results.

You will see the outputs of the program in the Result window. The Result window is also referred to as the "output console."

Reflection



Can you make any connections between the file name, parts of the code (statements) in the left window, and outputs in the results window on the right?

As you learn more about Java programming, make sure you record important learning concepts in your computer science notebook.

Computer Science Notebook: Refer to the [Computer Science Notebook](#) resource for more details.



Computer Science Notebook

Add to your notes: Sequencing defines an order for when steps in a process are completed. Steps in a sequence are completed one at a

time.

Compiler and Runtime Errors

Before the computer can run a Java program, it first uses a [compiler](#) to translate (or compile) the Java instructions into machine code. When you have a simple error related to semantics or syntax, the computer will output an error report that tells you what and where the problem was. A program with no compiler errors will compile successfully so it is ready to run.

A runtime error, on the other hand, happens when the program is running and tries to do something it is not supposed to or something unexpected. Maybe the program tries to use a piece of information that isn't there, or it tries to do a calculation that doesn't make sense. In this case, the program crashes and the computer reports a similar error.

Errors in programs are called [bugs](#), and fixing them is called [debugging](#).

Debug Your First Java Program

3

Examine the following code with compiler errors in it.



4

Run the program in the code editor. Describe what you see.

What did you see?

The results of the output console report that the file `HelloPLTW.java` has two errors in it.

In line 5, a semicolon ";" was expected at the end of the `System.out` statement.

In line 6, a curly brace "}" was expected at the end of the program.



EXPECT AN ERROR

Sometimes you will be given code with errors in it. By correcting the code, you will learn some of the rules of the Java programming language.

5

Based on the error report, try to correct the program by adding a semicolon (;) to end the Java statement and adding a close curly brace (}) to end the program.

6

Run the program to ensure both errors are fixed.



EXPECT AN ERROR

Now you have a new error. This new error is a runtime error. It did not show up initially when you ran the code because syntax errors are identified before runtime errors.

What did you see? —

Line number 1 has the error "class Hello is public, should be declared in a file named Hello.java."

7

Change the name of the class to match the name of the file HelloPLTW .

IMPORTANT: To create a program in Java, the class name has to be the same as the file name.

8

Run the program in the code editor. You should see one final error.



EXPECT AN ERROR

This is the final error you should see.

Need Help?

—

This error may be tough to spot. In Java, capitalization is very important. If you compare the very first program you ran successfully with this second buggy program, you should find one word that should not be capitalized.

9

Correct the final error and run the program to confirm all four errors were corrected.

Need Help?

—

The method name, `Main`, should not be capitalized.

Congratulations! You have successfully debugged your first Java program! Look at the table for the common errors you debugged.

Error	Description
Syntax Errors	The first two errors you encountered were <u>syntax</u> errors. These are very similar to grammatical errors in your English class, because the language elements of Java must be in the correct place.
Semicolon ;	All Java statements must end with a <u>semicolon</u> . Some Java statements can occupy more than one line in a program.
Curly Braces {} The error "reached end of file while parsing" means that there is a missing close-curly-brace } at the end of the file.	Curly braces are used to define blocks of code and code segments. The open-curly-brace { and close-curly-brace } are always used together; you cannot have an open curly brace without a matching close curly brace. The error "reached end of file while parsing" means that there is a missing close-curly-brace } at the end of the file.

NOTE: The notebook callout below indicates the work should be documented in your PLTW Computer Science Notebook.



Computer Science Notebook

Add to your notes:

- A compiler checks code for some errors. Errors detectable by the compiler need to be fixed before the program can be run.
 - A syntax error is a mistake in the program where the rules of the programming language are not followed. These errors are detected by the compiler.
-

The main Method

The header `public static void main(String[] args)` creates what is called the `main` method of the program. A **method** is a named block of code that only runs when it is called. A block of code is any section of code that is enclosed in braces, so the `main` method runs (executes) when you run the program. Stated another way, it is the entry point when you run a program.

ADDITIONAL INFORMATION: Java programs rely on a Java Virtual Machine (JVM) that allows the program to communicate with any computer. When a program is run, the JVM finds the `main` method and begins execution.

Review the parts of a Java program.

A semicolon ;

Should be at the end of every Java statement.

Curly braces { }

Signifies a block of code.

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

The main entry point of a
program.

```
public class  
className
```

Begins the Java program.

4 of 5

ClassName.java

The name of a Java file.

5 of 5



Computer Science Notebook

Add to your notes: A method is a named block of code that only runs when it is called. A block of code is any section of code that is enclosed in braces.

Printing and Escape Sequences

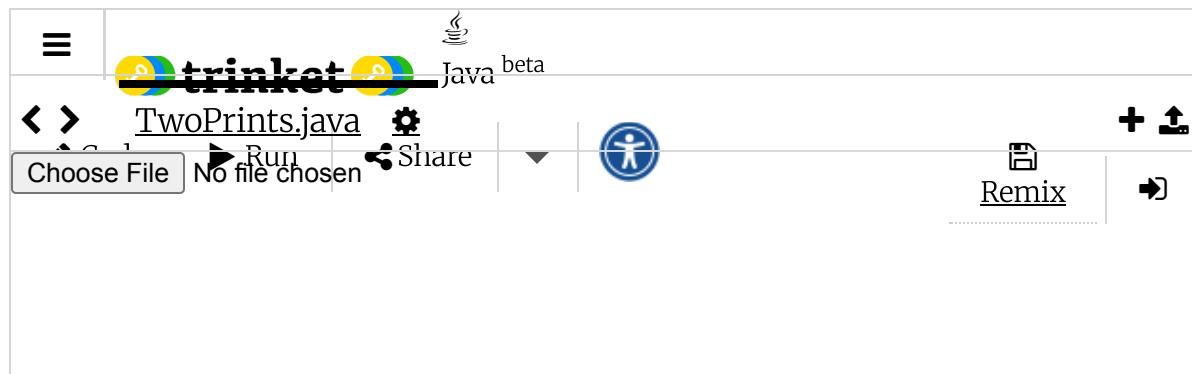
When programming, some concepts are so frequently represented that programmers can draw upon existing code that has already been tested, enabling them to write solutions more quickly and with a greater degree of confidence.

Output with `print` and `println`

You may have noticed the program you debugged used the `System.out.println` statement to generate output. You can generate different types of output using variations of the class `System.out`. In this course, you will learn two ways.

10

Examine the following code.



11

Change the first ellipses (three dots) to your first name.
Change the second ellipses (three dots) to your last name.



Computer Science Notebook

Add to your notes: A sequence of characters inside quotation marks, such as "...", is called a string literal. A literal is a representation of a fixed value.

12

In the code, duplicate each print statement multiple times ensuring each statement is on its own line. Can you determine what the difference in the outputs is between the `print()` and the `println()` statement?

13

Run the program. Notice how your names run together. Add a space in one of the string literals to fix this.

The `.println` version of the `System.out` object moves the cursor to a new line after the string literal has been displayed.

The `.print` version does not move the cursor to a new line. Each output appears right after the previous one instead of on a new line.

You can create a new line inside a print statement by adding `\n` anywhere inside the quotation marks. The following statements are equivalent:

```
System.out.print("Hello \n");
System.out.println("Hello ");
```

When used inside quotation marks, the `\` character is an escape character. It indicates that a special character follows. A special character preceded by a backslash `\` is an escape sequence.

For example, `\n` is an escape sequence that indicates the newline character. Used in a print statement, `\n` produces a new line of output.

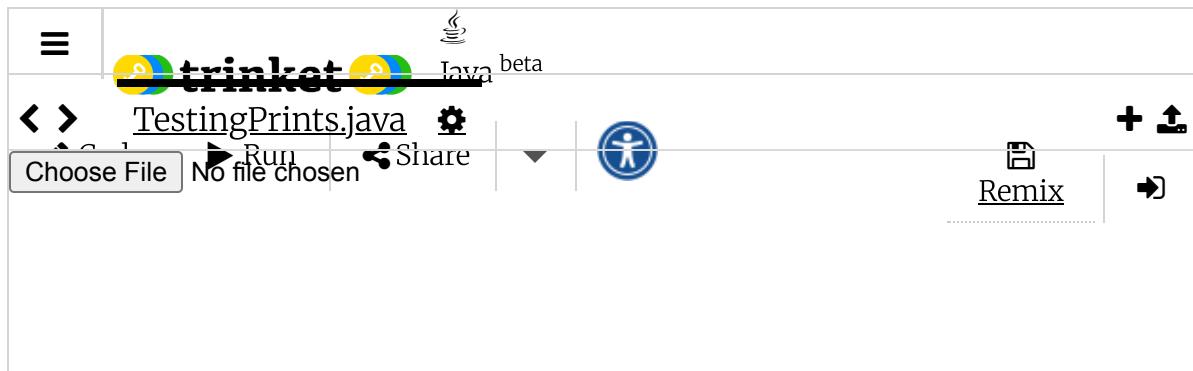


Computer Science Notebook

Add to your notes: Escape sequences such as `\\"`, `\\\`, and `\n` are special sequences of characters that can be included in a string. They start with a `\` and have a special meaning in Java.

14

Try it out! Can you use `print` statements to produce the same output as the `println` statements in the program below?



Other escape sequences are the double-quote `\"` and the backslash `\\\`. The escape sequence `\"` allows you to add a double quote to your program, and the `\\\` allows you to add a backslash character.

15

In `TestingPrints`, add a `print` statement to produce the following output, including the double

quotes.

"And I think to myself\what a wonderful
world"

Need Help?

—

```
System.out.print("\n\"And I think to myself\\what a wonderful  
world\"");
```

It's Your Turn

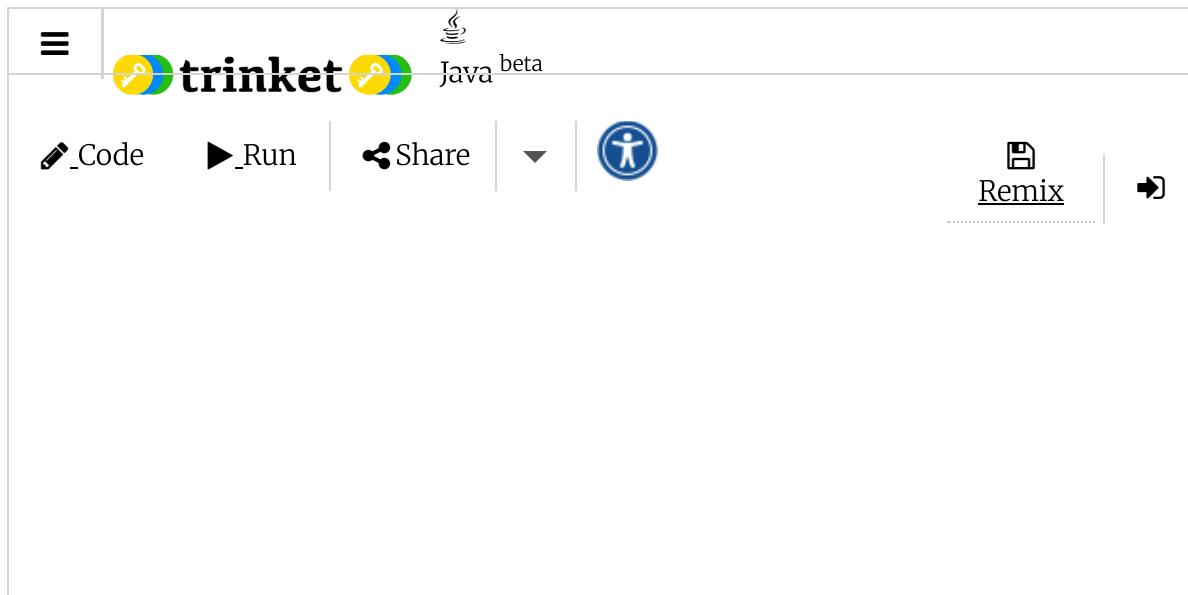
Now it's your turn to practice `print` and `println` methods, string literals, and escape sequences.

```
-----  
--0-----0--  
-----/ \----  
--\_____/-  
-----  
"Smile!"
```

16

Create the happy face image above using the following criteria:

- Use `System.out.print` at least two times.
- Use `System.out.println` at least two times.
- Use the three escape sequences you learned in this activity.



17

SCREENSHOT: Take a screenshot of your code and the output it produced.



Computer Science Notebook

It is very useful to document your errors as you learn about them so you can have a reference for when you encounter an error again. Record errors you encounter and summarize the Java syntax rules that fixes them.

Conclusion

Question 1

How are `System.out.print` and `System.out.println` different?

Question 2

Why is Java such a widely used language for development?

Question 3

Why is learning to program so important?