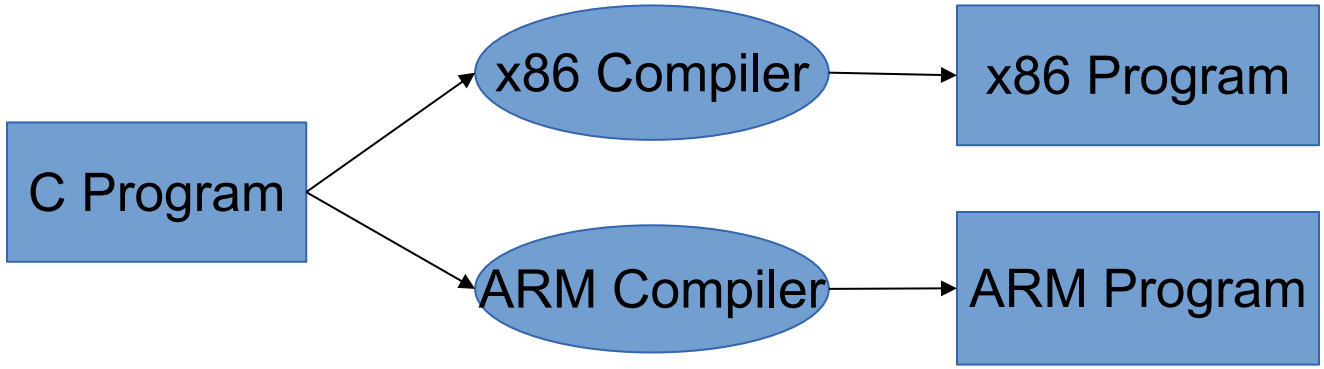
CS 149

Professor: Kevin Molloy
(adapted from slides originally developed by Alvin Chao)

JAVA is Portable

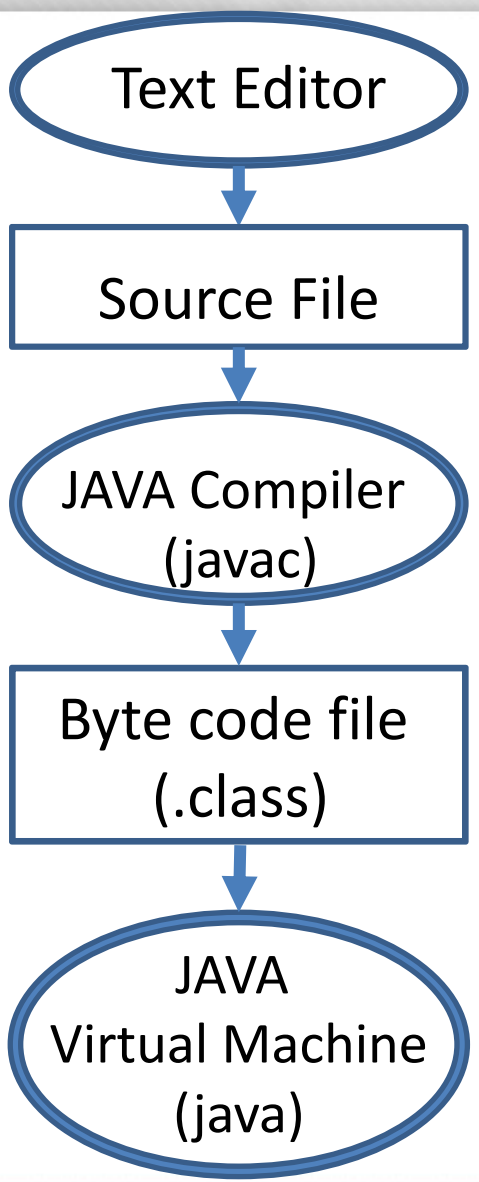
Most “high-level” languages are considered portable because they can be compiled into machine code for any computer:



Java Compilation

Byte Code Files are portable because there are JVM's that run on most machines

The **same** compiled byte code works on any JVM



Anatomy of a Java Program: Comments

Javadoc comments:

```
/**
 * Application that converts inches to centimeters.
 *
 * @author Chris Mayfield
 * @version 01/21/2014
 */
```

Everything between /** and */ ignored by compiler

Used to generate code documentation

Anatomy of a Java Program: Comments

Block comments are used for text that should *not* be part of the published documentation:

```
/*
   Permission is hereby granted, free of charge, to any
   person obtaining a copy of this software and associated
   documentation files (the "Software"), to deal in the
   Software without restriction.
*/
```

In-line comments are used for short clarifying statements:

```
// Create a scanner for standard input.
```

Anatomy of a Java Program: Classes

Java is an object-oriented language (OO)

- Java classes tie together instructions and data
- All Java code ***must*** exist within some class

```
public class ConvertInches {

}
```

`public` and `class` are keywords: Words that have a special meaning for Java.

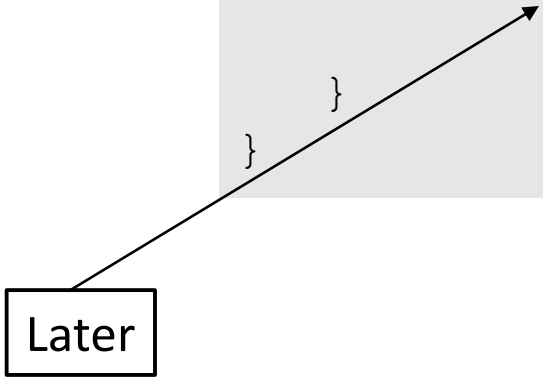
- `public` – (more later)
- `class` – Create a class with the following name. (Must match the file name)
  - Class names are always capitalized (by convention)

Braces { and } enclose blocks of code

Anatomy of a Java Program: Methods

Method – named collection of Java statements:

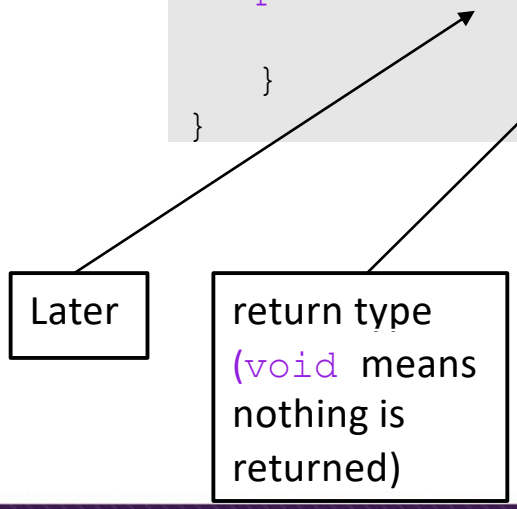
```
public class ConvertInches {

    public static void main(String[] args) {

    }
}
```



Anatomy of a Java Program: Methods

Method – named collection of Java statements:

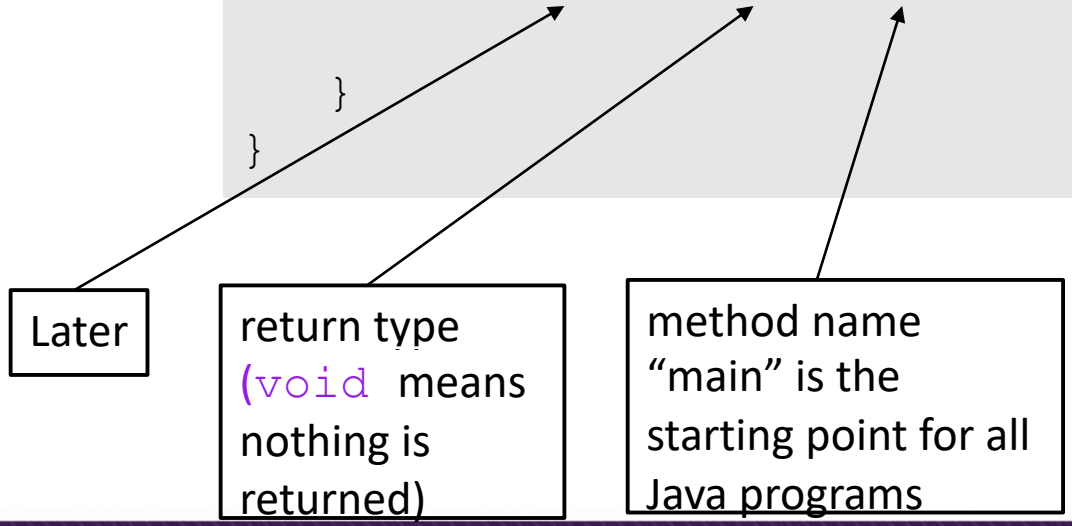
```
public class ConvertInches {

    public static void main(String[] args) {

    }
}
```



Anatomy of a Java Program: Methods

Method – named collection of Java statements:

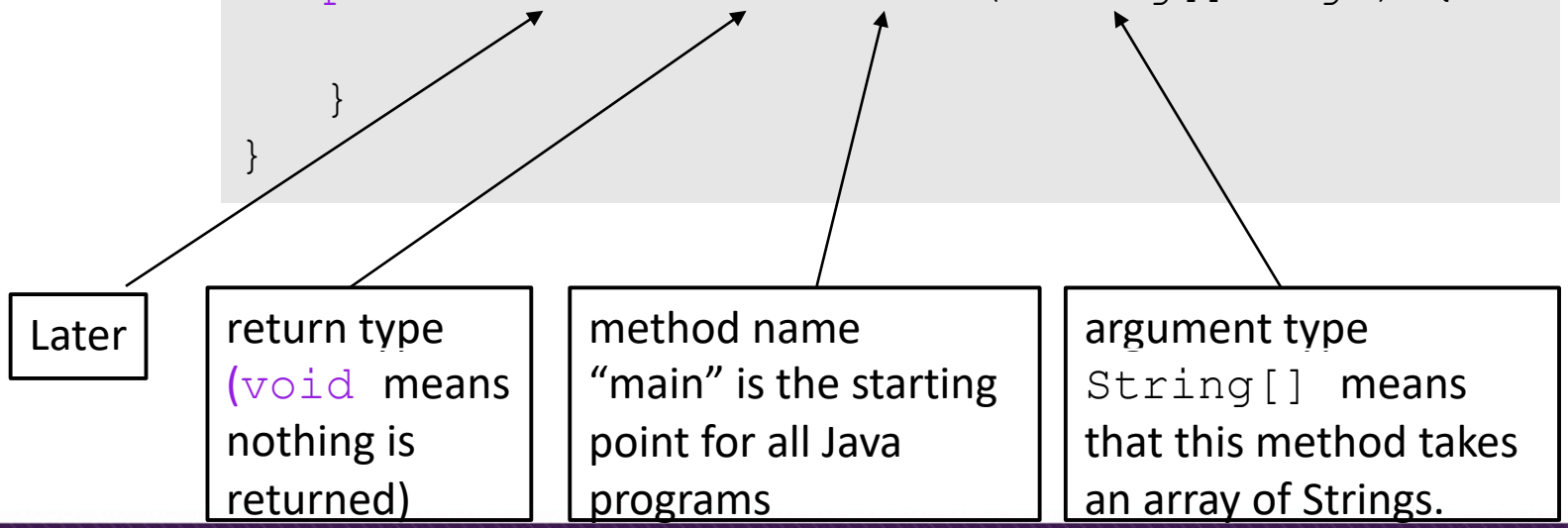
```
public class ConvertInches {

    public static void main(String[] args) {

    }
}
```



Anatomy of a Java Program: Methods

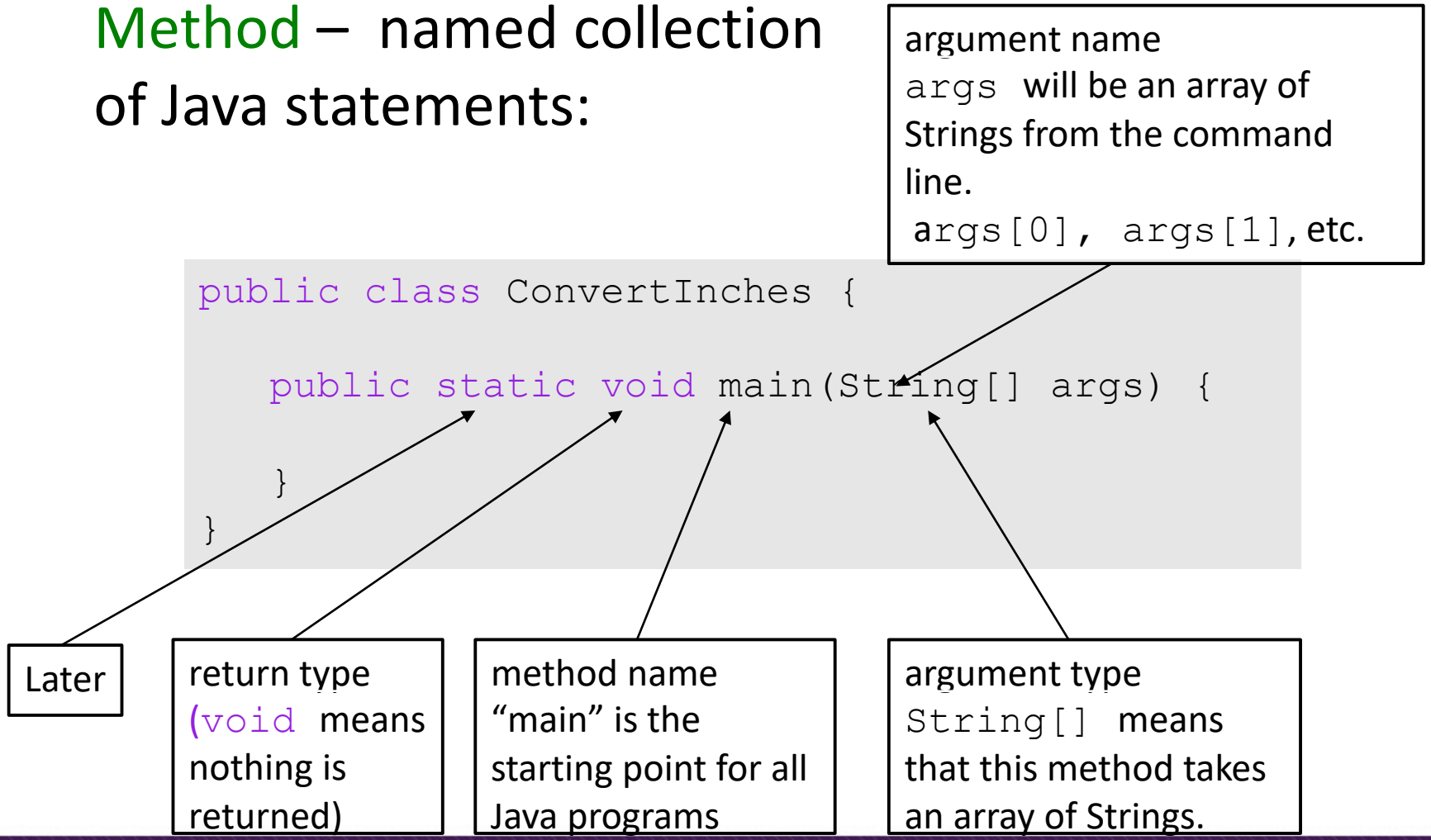
Method – named collection of Java statements:

```
public class ConvertInches {

    public static void main(String[] args) {

    }
}
```



Anatomy of a Java Program: Methods

Method – named collection of Java statements:



Anatomy of a Java Program: Declaring and Assigning Variables

variable – named box for storing data:

type

Defines what the variable can hold

name

Should always be informative. “x” is not OK.

```
int inch;
double cent;
final double CENT_PER_INCH;

CENT_PER_INCH = 2.54;
```

Anatomy of a Java Program: Declaring and Assigning Variables

variable – named box for storing data:

type

Defines what the variable can hold

name

Should always be informative. “x” is not OK.

```
int inch;
double cent;
final double CENT_PER_INCH;

CENT_PER_INCH = 2.54;
```

assignment

Puts the value on the right into the variable on the left.
ALWAYS RIGHT TO LEFT!

literal value

Anatomy of a Java Program: Declaring and Assigning Variables

variable – named box for storing data:

type

Defines what the variable can hold

name

Should always be informative. “x” is not OK.

`final`
makes this variable a constant

```
int inch;
double cent;
final double CENT_PER_INCH;

CENT_PER_INCH = 2.54;
```

assignment
Puts the value on the right into the variable on the left.
ALWAYS RIGHT TO LEFT!

literal value

Printing Output to the Screen

Two functions for printing:

- System.out.println()
- System.out.print()

What is the difference between these two functions?

Printing Output to the Screen

System.out.print(“Hello, world!\n”);	System.out.print(“Hello, world!\n”);

System.out.println includes the “newline” character.

Special characters are proceeded by a backslash (\) character (which is not printed).

- How do you print a backslash?
- How do you print a double quotation mark?

Printing Output to the Screen

Escape Sequence	Meaning
\\	Print a single backslash
\t	Print a tab character
\n	Print a newline character
\”	Print a double quotation mark