

CSCI 1304: Programming I

# Chapter 1 Introduction

## Part 2 A Sip of Java

Dr. Nour Almadhoun Alserr

Lecture 5

Week 3





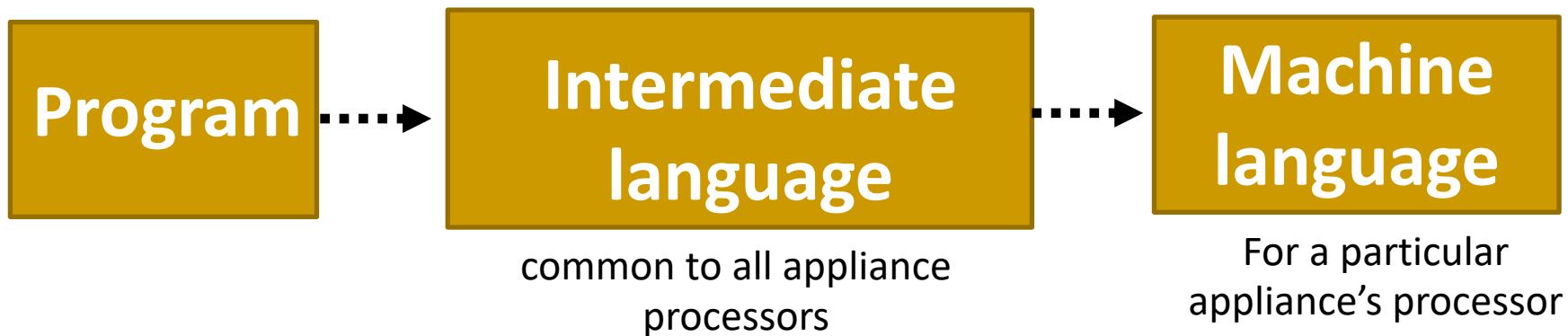
# Objectives

---

- Introduce program **design** and **errors**.
- **Overview** of the Java programming language.

# History of Java

- In 1991, James Gosling and Sun Microsystems began designing a language for **home appliances** (controlled by many **different** chips "processors").



# History of Java

- Appliance manufacturers weren't impressed ....
- In 1994, Gosling realized that his language would be ideal for a **Web browser** (over the Internet).



# Applications and Applets

---

- Two kinds of java programs: *applications* and *applets*
- **Applications**
  - Regular programs
  - Meant to be run on your computer
- **Applets**
  - Little applications
  - Meant to be sent to another location on the **internet** and run there (e.g., included with the HTML web document)

# First Java Application

```
import java.util.Scanner;
public class FirstProgram
{
    public static void main (String [] args)
    {
        System.out.println ("Hello out there.");
        System.out.println ("I will add two numbers for you.");
        System.out.println ("Enter two whole numbers on a line:");
        int n1, n2;
        Scanner keyboard = new Scanner (System.in);
        n1 = keyboard.nextInt ();
        n2 = keyboard.nextInt ();
        System.out.println ("The sum of those two numbers is");
        System.out.println (n1 + n2);
    }
}
```

# Some Terminology

- The person who writes a program is called the ***programmer***.
- The person who interacts with the program is called the ***user***.



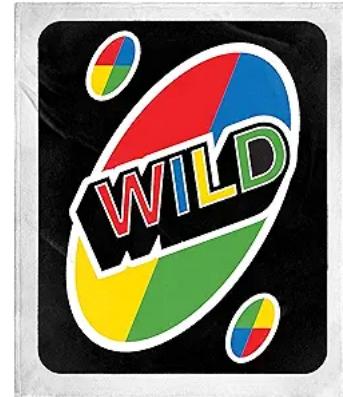
# Some Terminology

- A **package** is a **library of classes** that have been defined already.

□ **import java.util.Scanner;** //Explicit import.

- Scanner class is used to get user input.

□ **import java.util.\*;** //Wildcard import.



# Some Terminology

- A Java program can involve any number of classes.
- The class to run (**main method**) will contain the words
  - **public static void main(String args[])** { }

```
public void foo() {
    String [] args = new String[2];
    args[0] = "hello";
    args[1] = "every";

    System.out.println("Output: " + args[0] + args[1]);
}

// etc... the usage of 'args' here and in the main method is identical
}
```

# Some Terminology

- The **item(s)** inside parentheses are called **argument(s)** and provide the information needed by methods.
- A **variable** is something that can store data.
  - `int n1, n2;`
- An instruction to the computer is called a **statement**; it ends with a **semicolon**.
  - `System.out.println ("Hello out there.");`
  - `System.out.println ("I will add two numbers for you.");`
- The grammar rules for a programming language are called the **syntax** of the language.

# Printing to the Screen

□ **System.out.println ("Hello out there.");**

- **System.out** is an object for sending **output** to the **screen**.
- **println** is a **method** to print whatever is in parentheses to the screen.

# Printing to the Screen

## ■ Print() and println()

| Example   | Result              |
|---|---------------------|
| <pre>System.out.print("one");<br/>System.out.print("two");<br/>System.out.println("three");</pre>     | onetwothree         |
| <pre>System.out.println("one");<br/>System.out.println("two");<br/>System.out.println("three");</pre> | one<br>two<br>three |
| <pre>System.out.println();</pre>  | [new line]          |

# RECALL: First Java Application



```
import java.util.Scanner;
public class FirstProgram
{
    public static void main (String [] args)
    {
        System.out.println ("Hello out there.");
        System.out.println ("I will add two numbers for you.");
        System.out.println ("Enter two whole numbers on a line:");
        int n1, n2;
        Scanner keyboard = new Scanner (System.in);
        n1 = keyboard.nextInt ();
        n2 = keyboard.nextInt ();
        System.out.println ("The sum of those two numbers is");
        System.out.println (n1 + n2);
    }
}
```



# Printing to the Screen

- The object performs an action when you **invoke** or **call** one of its **methods**.

**objectName.methodName(argumentsTheMethodNeeds);**

- **Scanner keyboard = new Scanner (System.in);**
- **n1 = keyboard.nextInt ();**



# Compiling a Java Program or Class

---

- A Java program consists of one or more **classes**, which must be compiled before running the program.
- You need not compile classes that accompany Java (e.g. **System** and **Scanner**).



# Compiling a Java Program or Class

- Each class should be in a separate file.
- The name of the file should be the same as the name of the class.

```
public class FirstProgram  
{
```

# Compiling a Java Program or Class

## Java



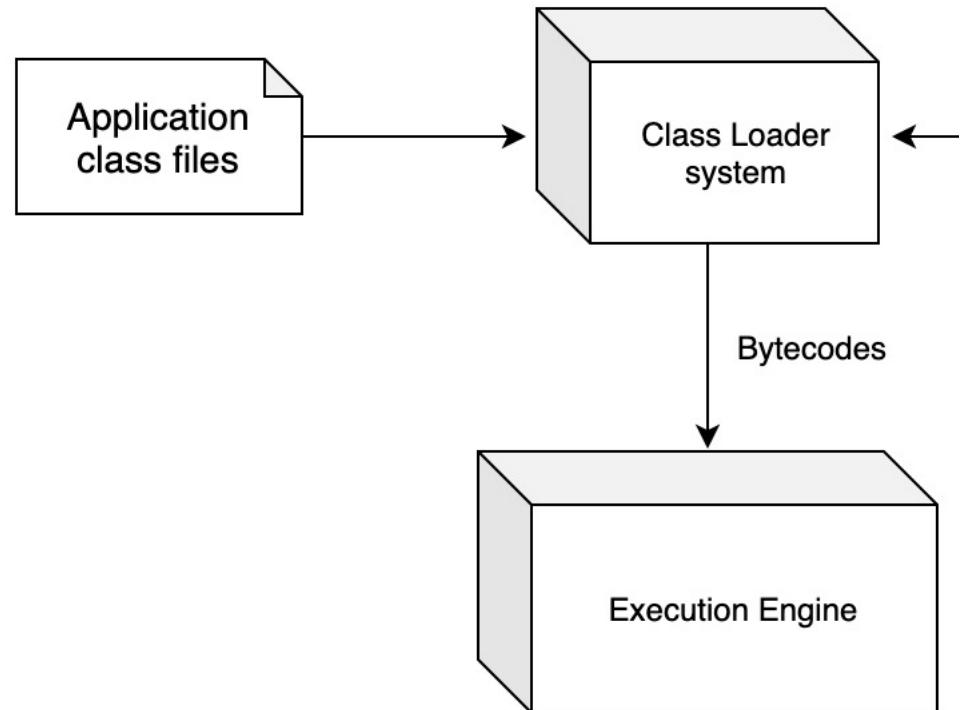
```
***** File name: Trial.java *****/
public class Geeks {
    public static void main(String[] args)
    {
        System.out.println("Hello world");
    }
}
```

```
javac Trial.java
Trial.java:9: error: class Geeks is public, should be
                     declared in a file named Geeks.java
public class Geeks
^
1 error
```

# RECALL: Class Loader



- A Java program typically consists of several pieces called **classes**.
- Each class may have a **separate author** and each is compiled (translated into byte-code) **separately**.



- A **class loader** (called a **linker** in other programming languages) automatically connects the classes together.

# Compiling and Running

- Use an **IDE (integrated development environment)** which combines a text editor with commands for compiling and running Java programs.

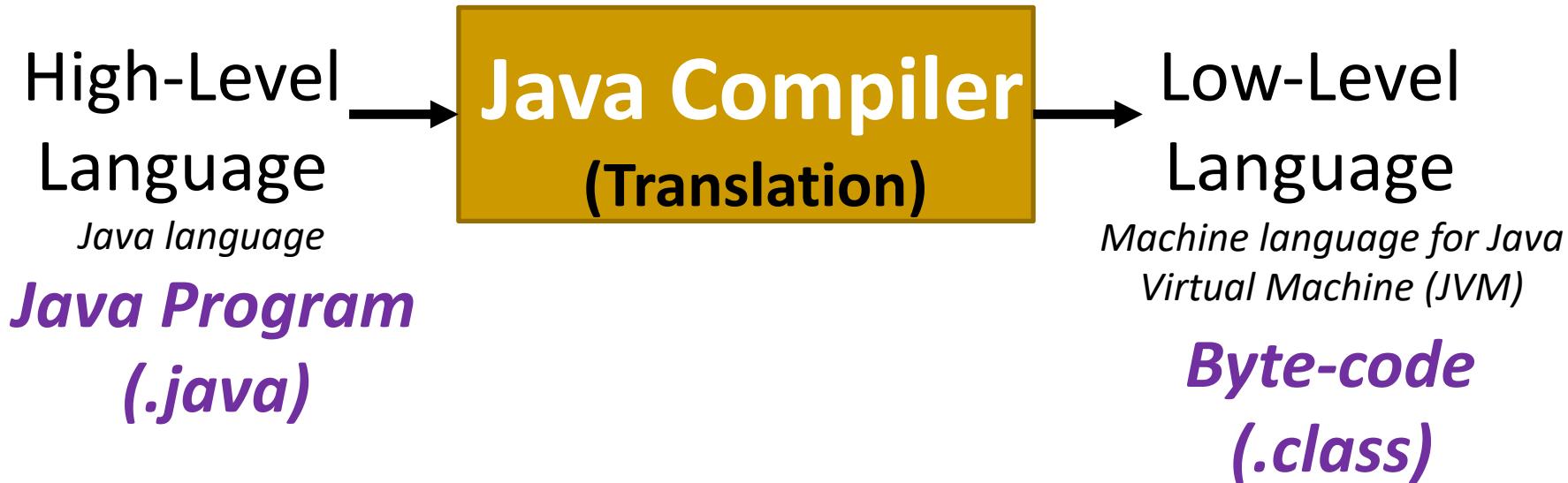
### Integrated development environment Software / java

From sources across the web

|   |  |  |
|---|--|--|
|  IntelliJ IDEA<br>Proprietary software |  Eclipse<br>Common Public License   |  NetBeans<br>Freeware   |
|  BlueJ<br>GNU General Public License   |  JDeveloper<br>Proprietary software |  JCreator<br>Freeware   |
|  jGRASP<br>Freeware                    |  MyEclipse<br>Proprietary software  |  DrJava<br>BSD licenses |

- When a Java program is compiled, the byte-code version of the program has the same name, but the ending is changed from **.java** to **.class**.

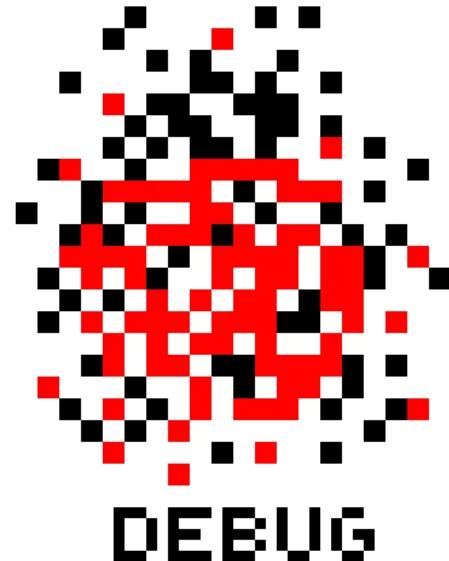
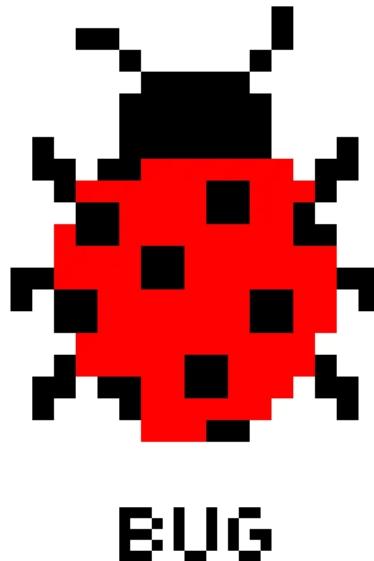
# RECALL: Java Compilers



- **Byte-code:** easy to translate into machine language for **any** particular computer (Java is platform **independent**).

# Errors

- An error in a program is called a **bug**.
- Eliminating errors is called **debugging**.



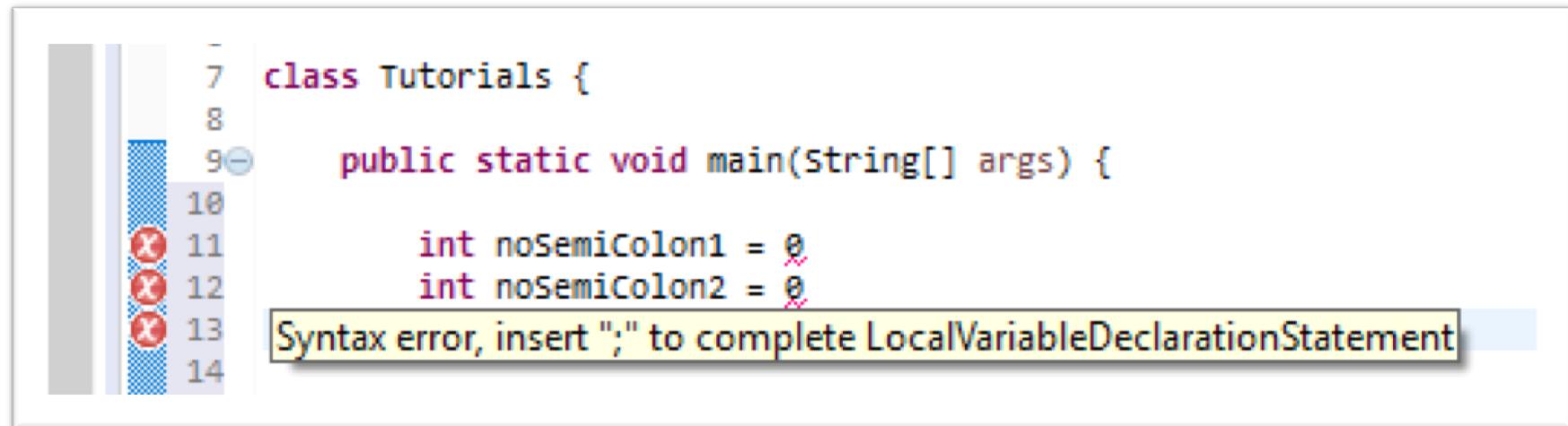
# Errors

---

- Three kinds or errors
  - Syntax errors
  - Runtime errors
  - Logic errors

# Syntax Error

- Grammatical mistakes in a program.
- The grammatical rules for writing a program are very strict.
- The compiler catches **syntax errors** and prints an error message.
- Example 1:



A screenshot of a Java code editor showing a syntax error. The code is as follows:

```
7  class Tutorials {  
8  
9      public static void main(String[] args) {  
10  
11          int noSemiColon1 = 0  
12          int noSemiColon2 = 0  
13  
14
```

The line "int noSemiColon1 = 0" has a red 'X' icon next to the number 11, indicating an error. A tooltip at the bottom right of the editor window says: "Syntax error, insert ";" to complete LocalVariableDeclarationStatement".

# Syntax Error

## Example 2:

Misspelled Keyword :

```
class SyntaxError {  
  
    public static void main(String[] args) {  
        int a = 2;  
        int b = 4;  
        system.out.println(a + b); // small s at start of System  
    }  
}
```

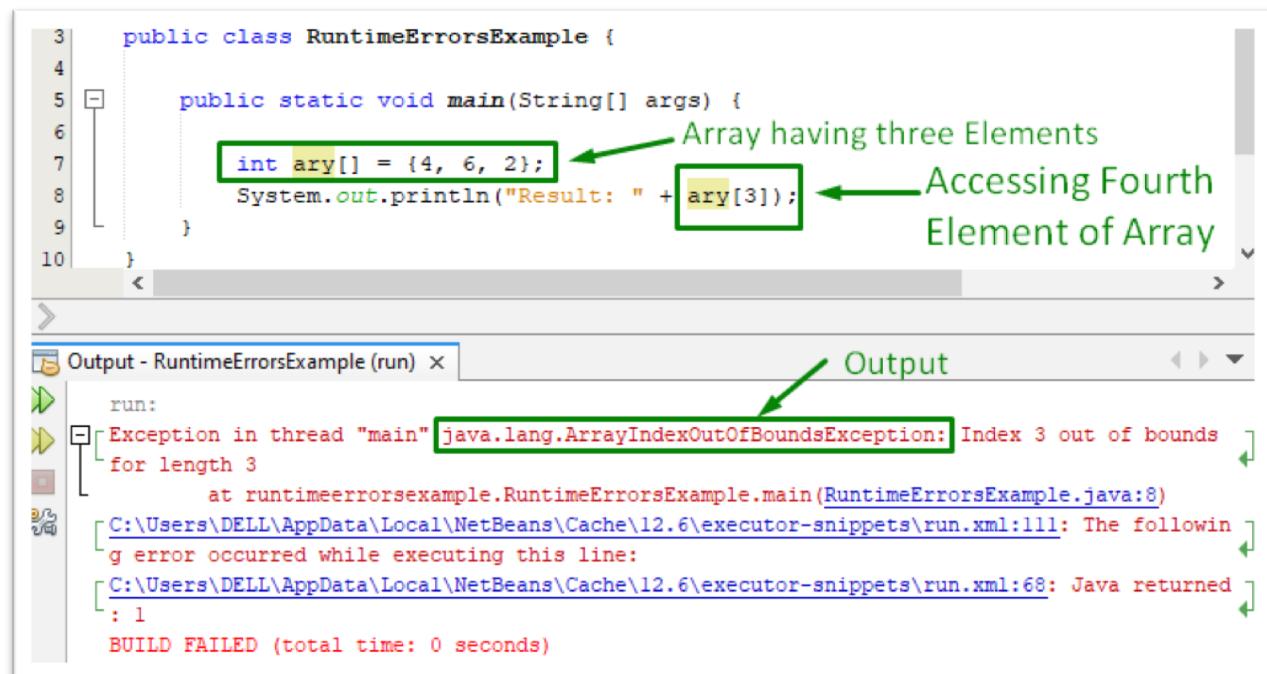
Example 1:

Output :

```
javac /tmp/UDRgGsLLQ6/SyntaxError.java  
/tmp/UDRgGsLLQ6/SyntaxError.java:5: error: package system does not exist  
system.out.println(a+b); // small s at start of System  
^  
1 error
```

# Runtime Errors

- Errors that are detected when your program is **running**, but not during compilation.
- When the computer detects an error, it **terminates** the program and prints an error message.
- Example 1:



The screenshot shows a Java code editor and an output window in NetBeans IDE.

**Java Code (RuntimeErrorsExample.java):**

```

3  public class RuntimeErrorsExample {
4
5      public static void main(String[] args) {
6
7          int ary[] = {4, 6, 2};
8          System.out.println("Result: " + ary[3]);
9
10 }

```

Annotations on the code:

- Line 7: `int ary[] = {4, 6, 2};` - Boxed as "Array having three Elements".
- Line 8: `System.out.println("Result: " + ary[3]);` - Boxed as "Accessing Fourth Element of Array".

**Output Window:**

Output - RuntimeErrorsExample (run) ×

```

run:
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds
for length 3
        at runtimeerrorsexample.RuntimeErrorsExample.main(RuntimeErrorsExample.java:8)
C:\Users\DELL\AppData\Local\NetBeans\Cache\12.6\executor-snippets\run.xml:111: The following
error occurred while executing this line:
C:\Users\DELL\AppData\Local\NetBeans\Cache\12.6\executor-snippets\run.xml:68: Java returned
: 1
BUILD FAILED (total time: 0 seconds)

```

Annotations on the output:

- Line 1: "Output" - Boxed as "Output".
- Line 2: `java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds` - Boxed as "Index 3 out of bounds".

# Runtime Errors

Example 2: Dividing an integer by zero :

```
class JavaError {  
  
    public static void main(String[] args) {  
        int a = 23;  
        int b = 0;  
  
        System.out.println(a / b);  
    }  
}
```

Output:

```
java -cp /tmp/UDRgGsLLQ6 JavaError  
Exception in thread "main" java.lang.ArithmetricException: / by ze
```

# Logic errors

- Errors that are not detected during compilation or while running, but which **cause the program to produce incorrect results.**
- Example: program to print even numbers.

```
class JavaError {  
  
    public static void main(String[] args) {  
        for (int i = 0; i <= 10; i++) {  
            if (i / 2 == 0) { // it should be % in place of /  
                System.out.println(i);  
            }  
        }  
    }  
}
```

# Logic errors

Actual Output :

```
0  
1
```



Expected Output :

```
0  
2  
4  
6  
8  
10
```



Do you have  
any  
QUESTIONS  
??

