

Introduction to programming

Why do we need programs?

- Humans communicate in a natural language
 - Large vocabulary (10 000s words)
 - Complex syntax
 - Semantic ambiguity
 - E.g., the man saw the boy holding his father's hand.
- Machines communicate in **binary code / machine language**
 - Small vocabulary (2 words... 1, 0)
 - Simple syntax
 - No semantic ambiguity

Translation of communication

Humans -- natural language

Large vocabulary
Complex syntax
Semantic ambiguity

Machines -- binary language

Small vocabulary
Simple syntax
No semantic ambiguity

Programming
(COMP 248 + COMP 249)

Programming language

Ex: ???

Vocabulary: restricted

Syntax: small and restricted

Semantic: no ambiguity (almost)

*Compiler +
Interpreter*

Origins of the Java Language

- Created by James Gosling at Sun Microsystems (1991), now owned by Oracle
- Originally designed for programming home appliances
- Its popularity has grown quickly since and it is pretty much everywhere (except home appliances)
- Is an **object-oriented programming (OOP)** language

Object Oriented Programming

- OPP treats everything in the world as objects (e.g., automobiles, universities, people)
- Each object has the ability to perform actions, and those actions can have an impact on other objects
- **OOP is a methodology that views a program as consisting of objects that interact with each other by means of actions**

Compilers

- A compiler is a software tool that **translates a high-level-language program** (e.g., Java program) into an equivalent **low-level-language program**
- In other languages (ex. C, C++) the compiler translates directly into machine language
 - But each type of CPU uses a different machine language ... so the same executable file will not work on different platforms
 - You will need to re-compile the original source code on different platforms
- **Java is different...**

Java Translation

- Java **compiler**:

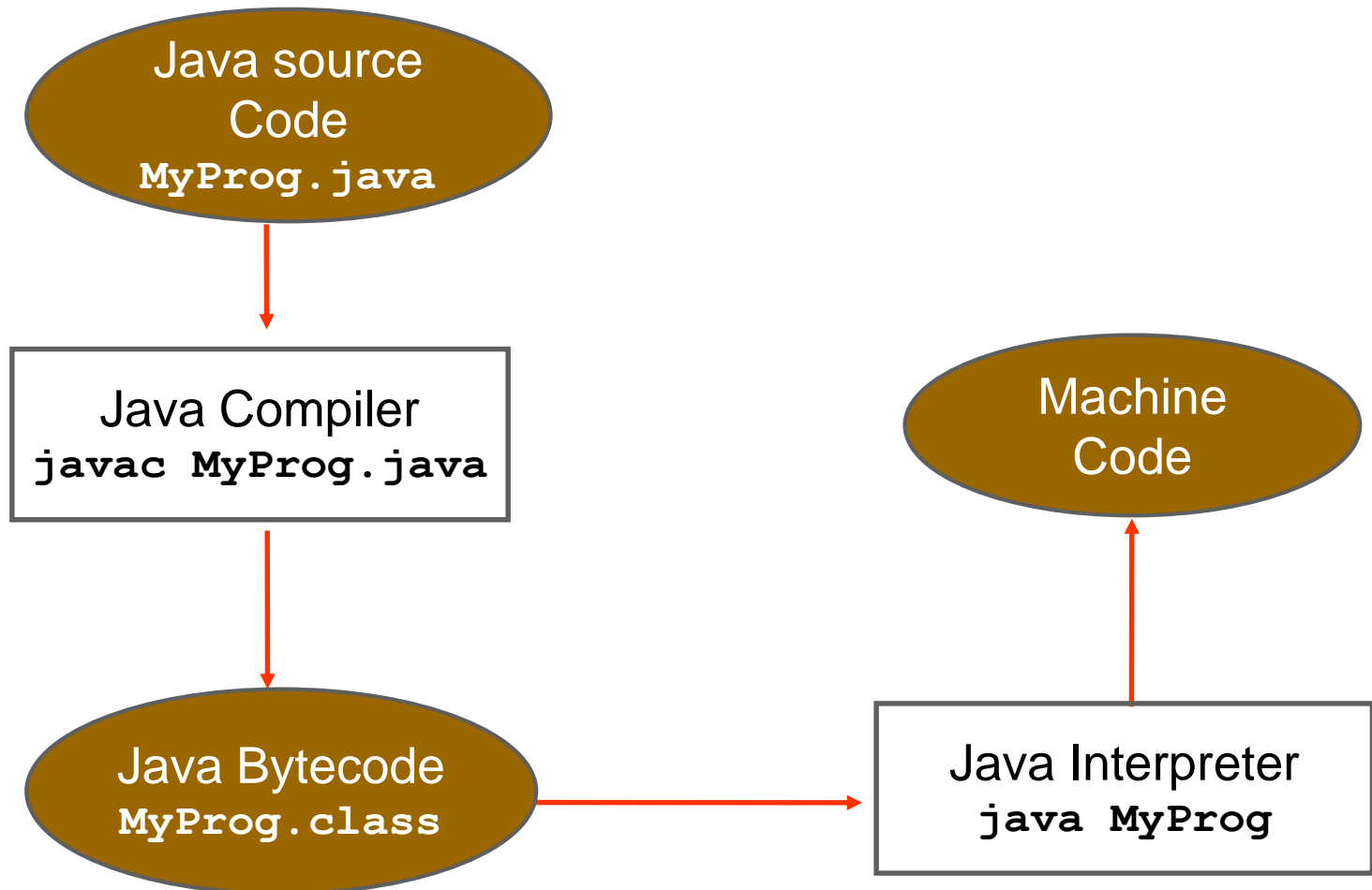
- Java source code --> **byte-code**
- a machine language for a fictitious computer called the *Java Virtual Machine (JVM)*

- Java **interpreter**:

- Translates Java byte-code into machine language and executes it
- Translating byte-code into machine code is relatively easy

- This way, the Java compiler is not tied to any particular machine
- Once compiled to *byte-code*, a Java program can be used on any computer, making it very portable

Java Translation



Some definitions

■ **Algorithm:**

a step-by-step process for solving a problem expressed in natural language

■ **Pseudocode:**

An algorithm expressed in a more formal code-like language, but does not necessarily follow a specific syntax

■ **Program:**

An algorithm expressed in a programming language that follows a specific syntax

Java Program Structure

- A java program:
 - is made up of one or more **classes** (*collection of actions*)
 - a class contains one or more **methods** (*action*)
 - a method contains program **statements/instructions**
- A Java program always contains a method called **main**

Java Program Structure

```
// comments about the class
```

```
public class MyProgram
```

```
{
```

```
}
```



class header



class body

MyProgram.java

Java Program Structure

```
// comments about the class
```

```
public class MyProgram
```

```
{
```

```
    // comments about the method
```

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

```
    {
```

```
    }
```

```
}
```

} method body

} method header

MyProgram.java

A small Java program

```
/** *****  
// Author: L. Kosseim  
//  
// Demonstrates the basic structure of a Java application.  
// *****  
  
public class Hello  
{  
    //-----  
    // Prints a message on the screen  
    //-----  
    public static void main (String[] args)  
    {  
        System.out.println ("Hello World!!!");  
    }  
}
```

— Hello.java



Java is case sensitive!

extension of java programs

You will type and run this program in tutorial 1

Syntax and Semantics

- **Syntax rules**

- define how we can put together symbols, reserved words, and identifiers to make a valid program

- **Semantics**

- define what a statement means

- A program that is syntactically correct is not necessarily logically (semantically) correct

Three types of errors

■ Compile-time (syntax) errors

- The compiler will find syntax errors and other basic problems
- An executable version of the program is not created

■ Run-time errors

- A problem can occur during program execution
- Causes the program to terminate abnormally

Three types of errors ...

- **Logical (semantic) errors**
 - A mistake in the algorithm
 - Compiler cannot catch them
 - A program may run, but produce incorrect results

Just checking...

The hardest kind of error to detect in a program is a:

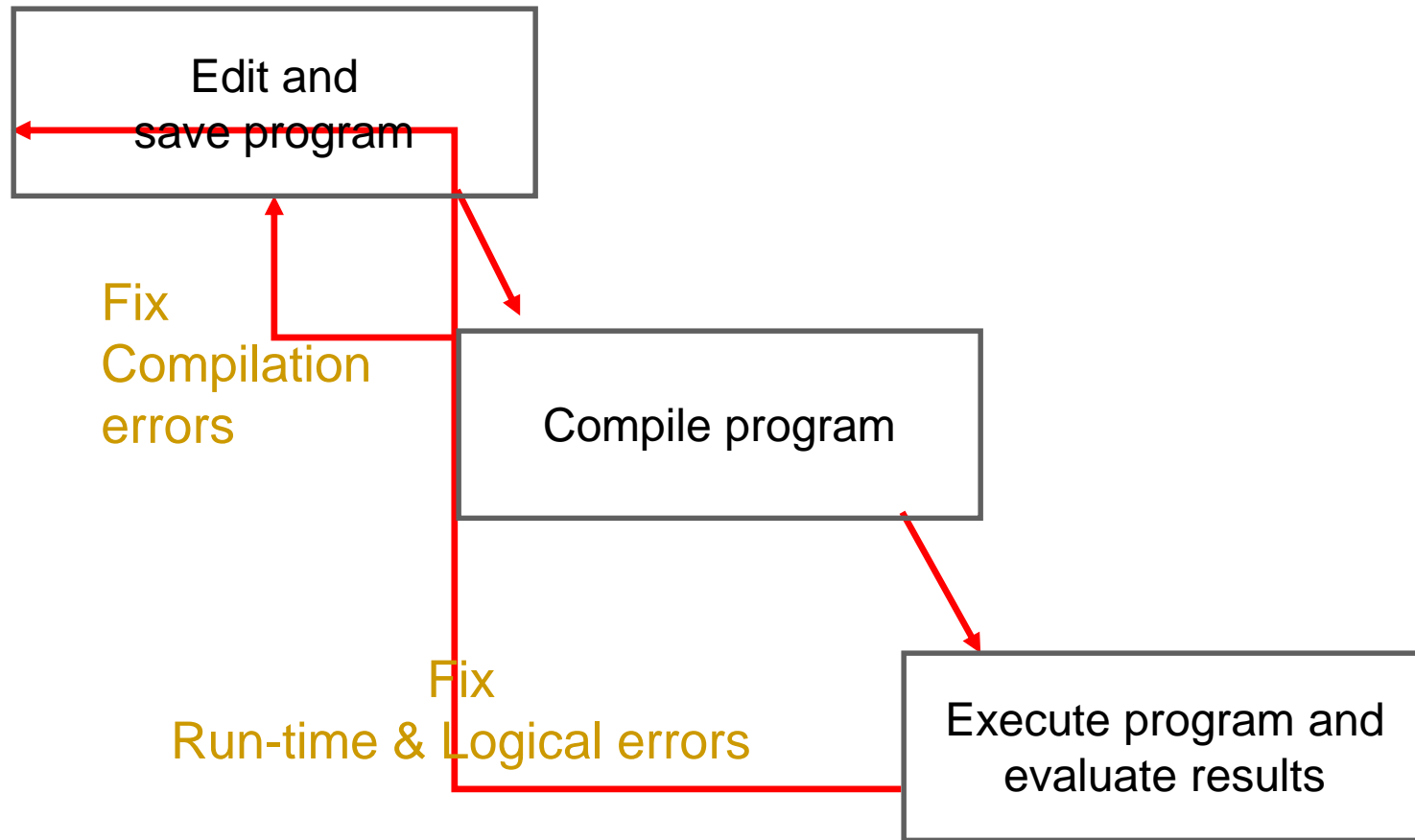
Syntax error

Run-time error

Logic error

All of the above

Basic Program Development



Development Environments

You need 2 things:

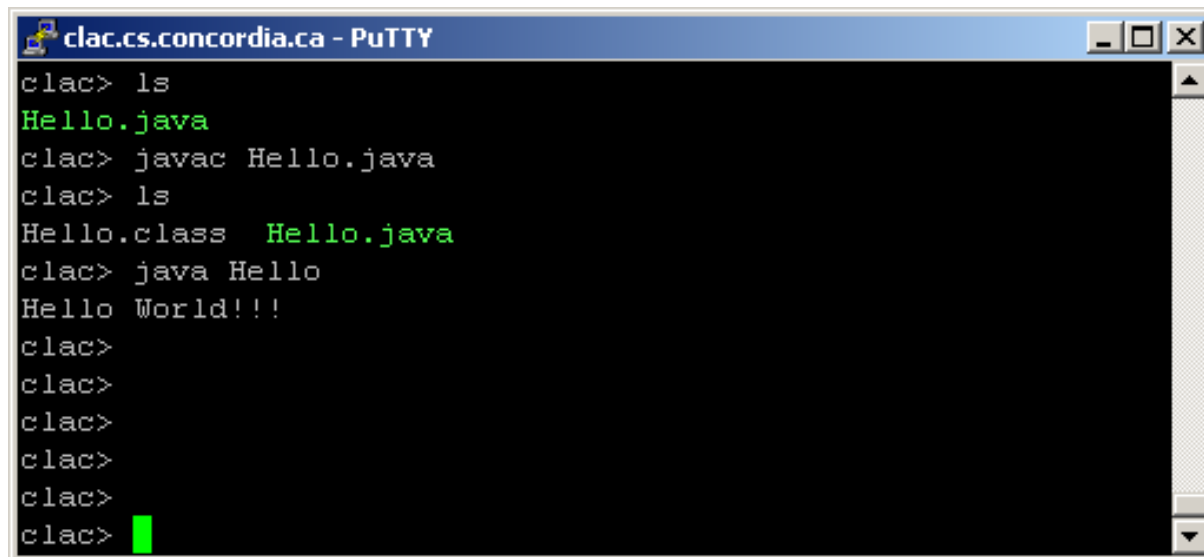
Basic compiler & interpreter

Java Development Kit (JDK) -- [download the SDK](#)

Compiler: `javac Hello.java`

The result is a byte-code program called: `Hello.class`

Interpreter: `java Hello`



```
clac.cs.concordia.ca - PuTTY
clac> ls
Hello.java
clac> javac Hello.java
clac> ls
Hello.class  Hello.java
clac> java Hello
Hello World!!!
clac>
clac>
clac>
clac>
clac>
clac>
clac>
```

Development Environments

- A text processor OR an IDE (Integrated Development Environment)
 - Eclipse
 - JCreator
 - Borland JBuilder
 - Microsoft Visual J++
 - ... see course Web site to download one
- In tutorial 1, you will edit, compile and run Hello.java

Types of Java programs

■ Applications

- "autonomous applications" or **stand-alone program** executed by the local OS (through the Java Virtual Machine)
- can use graphics and GUI or just plain console I/O what we will see in COMP 248

■ Applets

- "little **applications that have a windowing interface**" executed by a Web browser or an applet viewer (through the Java Virtual Machine)
- If used in a Web browser, it must be inserted into an HTML page
- must use a GUI

Next class

Java fundamentals