CS 241 — Introduction to Problem Solving and Programming

Fundamentals of Programming

A First Look at Java

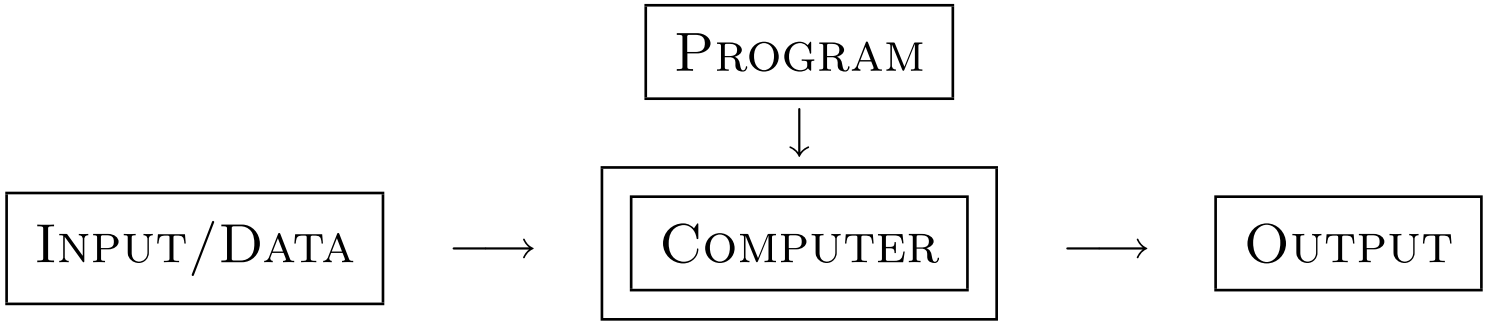
Jan 12 and 14, 2005

Outline/overview

- Programming languages
- A first loop at a program
- Variables
  - Declarations
  - Identifiers
  - Reserved words
  - Naming conventions

Stored Programs

- *Instructions* to a computer are stored in computer memory, as data is.
- The idea of a machine storing instructions is as old as the Jacquard loom (1801).
- A *program* is a set of instructions.



Computer instructions

```
11101001
00100100
10100100
01001011
01001010
10010010
10010010
01010010
10100101
10100100
01001011
01001010
```

Problem:

- Computer hardware accepts instructions only a language of extremely simple commands encoded in binary.
- Programming in the *machine language* is time-consuming and tedious.
- Machine languages are nearly incomprehensible to humans.
- Each type of computer architecture has its own language—programs are not *portable*.

High-level languages

Solution: *High-level languages.*

- HLLs should be relatively easy to learn and write in.
- They should be precise in their expression of logic and instructions.
- They must be translated or *compiled* to machine languages.
- They should be machine independent.

  Slogan for portability:

  Write once, compile anywhere.

Languages and Compilers

Programming languages include Pascal, COBOL, FORTRAN, C, C++, BASIC, Visual Basic, C#, Scheme, . . . and Java.

Traditionally, a *compiler* is a program that translates a program from a high level programming language (the *source*) to a machine language (the *target*).

(The idea of a compiler has been expanded to other programming language translation systems.)

History of programming languages leading to Java

- First compilers are completed: Fortran (1957), COBOL (1959).
- Simula, the first *object-oriented* programming language is developed (1960's).
- C is developed as a language in which to write Unix (1973).
- C++ is developed, extending C with object-oriented features.
- Java is developed as an alternative C-like object-oriented language (1995).
- Microsoft develops C# as a competitor to Java (2001).

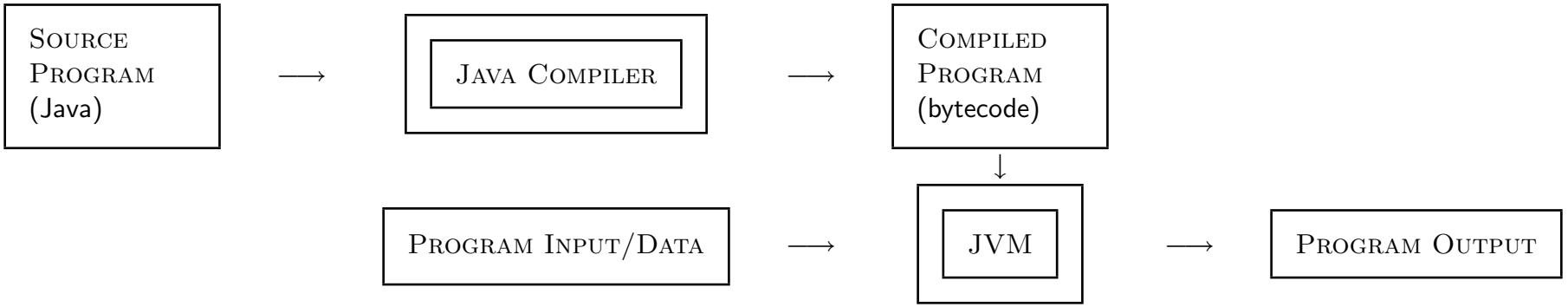
What's special about Java

- Originally designed for embedded devices.
- Quickly identified as a good web-based language and later general purpose language.
- Uses widely accepted C-style syntax.
- Implements object-oriented features better than C++ (debatable).
- Has gained popularity rapidly.
- Uses a new idea of portability.

The Java Virtual Machine

A *virtual machine* is a program that simulates a hypothetical computer.

Instead of compiling to a specific machine, the Java compiler compiles for the Java Virtual Machine.



A first Java program

```
public class FirstProgram {

    public static void main(String[] args) {

        System.out.println("Hello.");
        System.out.println("This is my first Java program.");
        System.out.println("Goodbye.");

    }
}
```

What's going on here?

A first Java program

```
public class FirstProgram {

    public static void main(String[] args) {

        System.out.println("Hello.");
        System.out.println("This is my first Java program.");
        System.out.println("Goodbye.");

    }
}
```

Ignore these parts for now. Note the name of the program, FirstProgram.

Dissecting a line

```
System.out.println("Hello.");
```

Take this as magic for now also.

Dissecting a line

```
System.out.println("Hello.");
```

We're telling Java to "print a line," or display a line on the screen.

Dissecting a line

```
System.out.println("Hello.");
```

What we want to display.

Dissecting a line

```
System.out.println("Hello.");
```

Finish the command.

A first Java program

The steps to run this are

- Save it to a file (`FirstProgram.java`)
- Compile it (using the Java Compiler, `javac`)
- Run it (using the Java Virtual Machine, `java`)

```
ar1121: {61} ls
FirstProgram.java
ar1121: {62} javac FirstProgram.java
ar1121: {63} ls
FirstProgram.class    FirstProgram.java
ar1121: {64} java FirstProgram
Hello.
This is my first Java program.
Goodbye.
ar1121: {65}
```

A first Java program

Now, what if we change it?

```
public class FirstProgram {

    public static void main(String[] args) {

        System.out.println("Hello.");
        System.out.println("This is my first Java program.");
        System.out.println("Goodbye.")

    }
}
```

What's different?

A first Java compiler error

```
ar1121: 112 javac FirstProgram.java
FirstProgram.java:8: ';' expected
^
1 error
```

The compiler rejects programs that don't conform to the language.

These are compiler errors or compile-time errors or syntax errors .

Variables

```
public class NoVariable {

    public static void main (String[] args) {

        System.out.println("Here is a number: 5");
        System.out.println("Here is the number again: 5");

    }
}
```

We can easily predict what this program will do.

```
Here is a number: 5
Here is the number again: 5
```

Variables

Now try this

```
public class Variable {

    public static void main (String[] args) {

        int number;
        number = 5;
        System.out.println("Here is a number: " + 5);
        System.out.println("Here is the number again: " + number);

    }
}
..
Here is a number: 5
Here is the number again: 5
```

Variables

What happened?

```
public class Variable {

    public static void main (String[] args) {

        int number;
        number = 5;
        System.out.println("Here is a number: " + 5);
        System.out.println("Here is the number again: " + number);

    }
}
```

- number is a variable .
- Variables are used to store a value in memory.
- Contrast with a constant or literal , such as 5.

Variables

The declaration of a variable:

```
int number;
```

The assignment of a variable:

```
number = 5;
```

- Variables must be declared before they are used in any way.
- The first assignment is called an initialization.
- Variables must be initialized before they are used in another way.

Errors involving variables

```
public class Variable {

    public static void main (String[] args) {

        number = 5;
        System.out.println(number);

    }
}
...
ar1121: 137 javac Variable.java
Variable.java:6: cannot resolve symbol
symbol  : variable number
location: class Variable
        number = 5;
        ^
Variable.java:7: cannot resolve symbol
symbol  : variable number
location: class Variable
        System.out.println(number);
                           ^
2 errors
```

```
public class Variable {

    public static void main (String[] args) {

        int number;
        System.out.println(number);

    }
}
...
ar1121: 138 javac Variable.java
Variable.java:7: variable number might not have been initialize
        System.out.println(number);
                           ^
1 error
```

Variables

Variables are so called because they can vary. We can reassign them.

```
public class Variable {

    public static void main (String[] args) {

        int number;
        number = 5;
        System.out.println("Here is a number: " + 5);
        number = 6;
        System.out.println("Here is another number: " + number);

    }
}
```

Make sure you understand *commands in a program are executed in order.*

Identifiers

Java has rules about naming variables. A name given to a variable (or other entities we will name) is an identifier .

Identifiers may contain

- letters (case sensitive)
- digits (except for the first character)
- underscores

Valid identifiers:

```
hello
Hello3
_hello
helloAndGoodbye
hello_and_goodbye
```

Invalid identifiers:

```
3hello
h@e&&ll%o
```

Reserved words

Certain words out off limits as identifiers because they have a predefined meaning in the language. These are called keywords or reserved words .

abstract	default	if	private	this
boolean	do	implements	protected	throw
break	double	import	public	throws
byte	else	instanceof	return	transient
case	extends	int	short	try
catch	final	interface	static	void
char	finally	long	strictfp	volatile
class	float	native	super	while
const	for	new	switch	
continue	goto	package	synchronized	

Naming conventions

Naming conventions have been accepted for Java (and are required for this class):

- The first letter should be lowercase.
- All other letters should be lower case except when using internal capitalization to join words, e.g. `averageCost`.
- Avoid digits.
- Do not use underscores.

But even more importantly, choose identifiers that are meaningful .

Summary

Be able to identify the following concepts:

- Program
- Compiler
- Machine language
- High level language
- Bytecode
- Virtual machine
- `System.out.println`
- Syntax error
- Variable
- Literal
- Identifier
- Declaration
- Assignment
- Initialization
- Reserved words