Chapter 1: Introduction

Java Programming Language
How the Java Virtual Machine Works (compiling, etc…)

Update by: Zoran Duric
THE JAVA PROGRAMMING LANGUAGE

Coming up: Java
Java

- The Java programming language was created by Sun Microsystems, Inc.
- It was introduced in 1995 and its popularity has grown quickly since.
- Java is NOT Javascript… they are different languages… knowing one does NOT mean you know the other! Don’t get that confused.

Coming up: Java Program Structure
In the Java programming language:

- A program is made up of one or more *classes*
- A class contains one or more *methods*
- A method contains program *statements*

- A Java application always contains a method called *main*
- *main* is the first method called when a Java application starts
Java Program Structure

// comments about the class

public class MyProgram
{

class header

class body

Comments can be placed almost anywhere

}

Coming up: Java Program Structure
public class MyProgram
{
    // comments about the method
    public static void main (String[] args)
    {
        method body
    }
}
• Time to write HelloWorld in Java

Coming up: In Java, what is used to separate blocks?
In Java, what is used to separate blocks?

- A. white space
- B. Tabs
- C. curly braces --- `{ }`
- D. square braces --- `[ ]`
Comments

- Comments in a program are called *inline documentation*
- They should be included to explain the purpose of the program and describe processing steps
- They do not affect how a program works
- Java comments can take three forms:
  
  ```java
  // this comment runs to the end of the line
  
  /*  this comment runs to the terminating symbol, even across line breaks  */
  
  /** this is a javadoc comment  */
  ```

Coming up: Identifiers
Identifiers

- **Identifiers** are the words a programmer uses in a program (variable names, class names, method names…)

- An identifier can be made up of letters, digits, the underscore character ( _ ), and the dollar sign.

- Identifiers **cannot** begin with a digit.

- **Java is case sensitive** - **Total**, **total**, and **TOTAL** are **different** identifiers.

- By convention, programmers use different case styles for different types of identifiers, such as:
  - *title case for class names* - **Lincoln**
  - *upper case for constants* - **MAXIMUM**

Coming up: Reserved Words are predefined in the language, and cannot be used as identifiers.
Reserved Words are predefined in the language, and cannot be used as identifiers

The Java reserved words:

<table>
<thead>
<tr>
<th>abstract</th>
<th>else</th>
<th>interface</th>
<th>switch</th>
</tr>
</thead>
<tbody>
<tr>
<td>assert</td>
<td>enum</td>
<td>long</td>
<td>synchronized</td>
</tr>
<tr>
<td>boolean</td>
<td>extends</td>
<td>native</td>
<td>this</td>
</tr>
<tr>
<td>break</td>
<td>false</td>
<td>new</td>
<td>throw</td>
</tr>
<tr>
<td>byte</td>
<td>final</td>
<td>null</td>
<td>throws</td>
</tr>
<tr>
<td>case</td>
<td>finally</td>
<td>package</td>
<td>transient</td>
</tr>
<tr>
<td>catch</td>
<td>float</td>
<td>private</td>
<td>true</td>
</tr>
<tr>
<td>char</td>
<td>for</td>
<td>protected</td>
<td>try</td>
</tr>
<tr>
<td>class</td>
<td>goto</td>
<td>public</td>
<td>void</td>
</tr>
<tr>
<td>const</td>
<td>if</td>
<td>return</td>
<td>volatile</td>
</tr>
<tr>
<td>continue</td>
<td>implements</td>
<td>short</td>
<td>while</td>
</tr>
<tr>
<td>default</td>
<td>import</td>
<td>static</td>
<td></td>
</tr>
<tr>
<td>do</td>
<td>instanceof</td>
<td>strictfp</td>
<td></td>
</tr>
<tr>
<td>double</td>
<td>int</td>
<td>super</td>
<td></td>
</tr>
</tbody>
</table>
public void test() {
    int myVar = 10;
    int MYVAR = 20;
    myVar = 35;
    System.out.println("Value is:"+MYVAR);
}

A. 10  B. 20  C. 35  D. MYVAR
White Space

- Spaces, blank lines, and tabs are called white space.
- White space is used to separate words and symbols in a program.
- Extra white space is ignored in Java.
- A valid Java program can be formatted many ways.
- Programs should be formatted to enhance readability, using consistent indentation.

- See HelloWorld2.java
- See HelloWorld3.java

Coming up: Object-Oriented Programming
Object-Oriented Programming

- Java is an object-oriented programming language
- As the term implies, an object is a fundamental entity in a Java program
- Objects can be used effectively to represent real-world entities
- For instance, an object might represent a particular employee in a company
- Each employee object handles the processing and data management related to that employee

Coming up: Objects
Objects

- An object has:
  - *state* - descriptive characteristics
  - *behaviors* - what it can do (or what can be done to it)

- The state of a bank account includes its account number and its current balance

- The behaviors associated with a bank account include the ability to make deposits and withdrawals

- Note that the behavior of an object might change its state
Classes

- An object is defined by a class
- A class is the blueprint of an object
- The class uses methods to define the behaviors of the object
- The class that contains the main method of a Java program represents the entire program
- A class represents a concept, and an object represents the embodiment of that concept
- Multiple objects can be created from the same class
• Let's create the BankAccount class
Coming up: Given Bill Gates' bank account and Dan Fleck's bank account, is "withdraw" a valid behavior in both?

Instances of a class have the state filled in. In this case, state is the name of the account holder and the balance.
Given Bill Gates’ bank account and Zoran Duric’s bank account, is “withdraw” a valid behavior in both?

- A. Yes
- B. No

If the answer is Yes, then all the behaviors between Bill Gates’ account and ZD’s account are the same... what then is different? (Text Answer)
If I give you a “Computer” and a “Dell Inspiron Laptop” which would be the class and which would be the object?

- A. Dell is Class, Computer is Object
- B. Both are Classes
- C. Both are Objects
- D. Dell is Object, Computer is Class

Coming up: Inheritance
Inheritance

- One class can be used to derive another via inheritance
- Classes can be organized into hierarchies

Coming up: Creating an object and calling a method
Creating an object and calling a method

- To construct an object of class, use the “new” keyword

- Rectangle myRect = new Rectangle();
  - This is similar to declaring a primitive variable, but you must call “new classname”

- Call a method on myRect:
  - width = myRect.getWidth();

Coming up: How do you know what methods you can call?
How do you know what methods you can call?

- Let's look at the Java API!

- [http://java.sun.com/javase/6/docs/api/](http://java.sun.com/javase/6/docs/api/)
HOW JAVA COMPILING WORKS
Program Development

- The mechanics of developing a program include several activities
  - writing the program in a specific programming language (such as Java)
  - translating the program into a form that the computer can execute
  - investigating and fixing various types of errors that can occur

- Software tools can be used to help with all parts of this process

Coming up: Language Levels
Language Levels

- There are four programming language levels:
  - machine language
  - assembly language
  - high-level language
  - fourth-generation language

- Each type of CPU has its own specific *machine language*

- The other levels were created to make it easier for a human being to read and write programs

Coming up: Programming Languages
Each type of CPU executes only a particular *machine language*. A program must be translated into machine language before it can be executed. A *compiler* is a software tool which translates *source code* into a specific target language. Often, that target language is the machine language for a particular CPU type. The Java approach is somewhat different.
Programming Languages

- Star Trek

Coming up: Programming Languages
Programming Languages

- C/C++ (traditional)
Programming Languages

- C/C++ (traditional)

C Program

Windows Architecture Compiler

MAC Architecture Compiler

Unix Architecture Compiler

Windows Machine Code

MAC Machine Code

Unix Machine Code

Coming up: Java Translation
Java Translation

- The Java compiler translates Java source code into a special representation called *bytecode*.

- Java bytecode is not the machine language for any traditional CPU.

- Another software tool, called an *interpreter*, translates bytecode into machine language and executes it.

- Therefore the Java compiler is not tied to any particular machine.

- Java is considered to be *architecture-neutral*.

Coming up: Java Bytecode
Java Bytecode

- Using Bytecode allows Java to be considered “write once, run anywhere”.

- The only code that must be specific for the platform is the Java Virtual Machine (JVM).

- This is why for different computers you need to download the appropriate JVM, but then any compiled Java code will run there.

Coming up: How TO USE THE Java compiler
So, if I have to eventually get to machine code either way using an architecture specific compiler (for C) or a specific VM (for Java), what is the benefit of Java’s bytecode -> Interpreter-> Machine code architecture?
HOW TO USE THE JAVA COMPILER
How to compile

• Install the Java Development Kit (JDK). This includes the java compiler (javac) and the JRE (Java Runtime Environment)

• Compiling:
  javac <<source file>>.java
  java HelloWorld.java

• Running:
  java <<classfile>>
  java HelloWorld

Coming up: Where to get Java
Where to get Java

- http://java.sun.com
- Click on JSE  (Java Standard Edition)
- (We will be using the Netbeans Integrated Development Environment (IDE) in class)