TRINITY COLLEGE FOR WOMEN NAMAKKAL

Department of Computer Science

JAVA PROGRAMMING

19UCS09-EVEN Semester

Presented by

M.MALATHI

Assistant Professor

Department of Computer Science

http://www.trinitycollegenkl.edu.in/

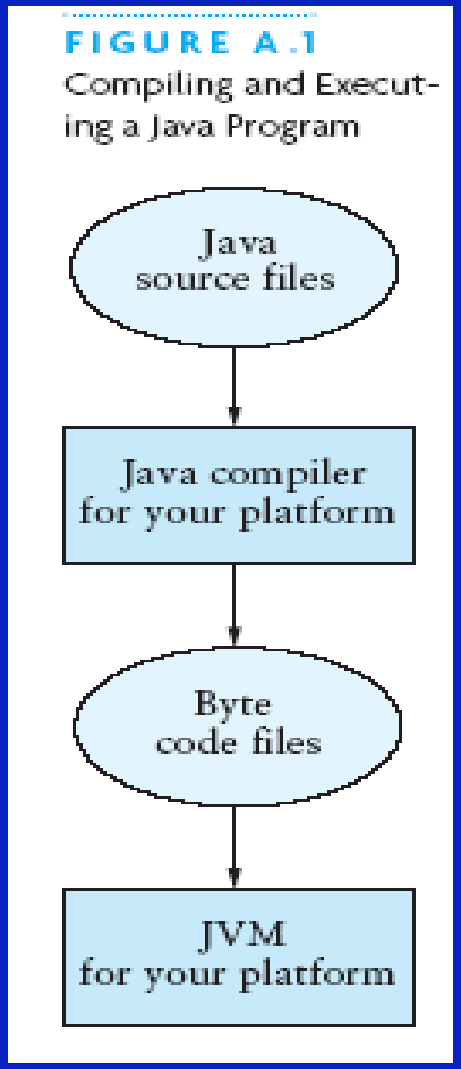
Characteristics of Java

- Java is ***platform independent:*** the same program can run on any correctly implemented Java system
- Java is ***object-oriented:***
  - ✓ Structured in terms of ***classes***, which group data with operations on that data
  - ✓ Can construct new classes by ***extending*** existing ones
- Java designed as
  - ✓ A ***core language*** plus
  - ✓ A rich collection of ***commonly available packages***
- Java can be embedded in Web pages

Java Processing and Execution

- Begin with Java ***source code*** in text files: **Model.java**
- A Java source code compiler produces Java ***byte code***
  - Outputs one file per class: **Model.class**
  - May be standalone or part of an IDE
- A ***Java Virtual Machine*** loads and executes class files
  - May compile them to native code (e.g., x86) internally

Compiling and Executing a Java Program



Classes and Objects

- The ***class*** is the unit of programming
- A Java program is a ***collection of classes***
  - Each class definition (usually) in its own **.java** file
  - *The file name must match the class name*
- A class describes ***objects (instances)***
  - Describes their common characteristics: is a *blueprint*
  - Thus all the instances have these same characteristics
- These characteristics are:
  - ***Data fields*** for each object
  - ***Methods*** (operations) that do work on the objects

Grouping Classes: The Java API

- API = *Application Programming Interface*
- Java = small core + extensive collection of packages
- A ***package*** consists of some related Java classes:
  - Swing: a GUI (graphical user interface) package
  - AWT: Application Window Toolkit (more GUI)
  - util: utility data structures (important to CS 187!)
- The ***import*** statement tells the compiler to make available classes and methods of another package
- A ***main*** method indicates where to begin executing a class (if it is designed to be run as a program)

A Little Example of **import** and **main**

```
import java.io.*;
    // all classes from javax.swing
public class HelloWorld {  // starts a
class
  public static void main (String[]
args) {
  // starts a main method
  // in: array of String; out: none (void)
  }
}
```

public = can be seen from any package
static = not “part of” an object

Processing and Running **HelloWorld**

javac HelloWorld.java

Produces **HelloWorld.class** (byte code)

java HelloWorld

Starts the JVM and runs the **main** method

Example:

```
class FibonacciExample1{
public static void main(String args[])
{
 int n1=0,n2=1,n3,i,count=10;
 System.out.print(n1+" "+n2);//printing 0 and 1

 for(i=2;i<count;++i)//loop starts from 2 because 0 and 1 are a
lready printed
 {
  n3=n1+n2;
  System.out.print(" "+n3);
  n1=n2;
  n2=n3;
 }

}}
```

THANK YOU

http://www.trinitycollegenkl.edu.in/