BCS 345 Java Programming

Arthur Hoskey, Ph.D.
Farmingdale State College
Computer Systems Department

- First Java Program
- Compilation and Running Sequence

Today's Lecture

Commenting and Indenting

- ***Commenting***
  ALL programs must be properly commented!

- ***Indenting***
  ALL programs must be properly indented!

- Description of how to properly comment and indent is in document on Blackboard.

Important Java Rules

- Java is case sensitive (same as C++).
- Most lines end in a semicolon.
- Most code is placed inside of a class.
- The name of a file should be the same as the name of the class inside of the file.

First Java Program

```
// IMPORTANT – The file should be named Welcome1 since
// the class it contains is named Welcome1 (case sensitive)

// Should be stored in file named Welcome1.java

public class Welcome1
{

   public static void main( String args[] )
   {
         System.out.printf("Hello World\n");

         System.out.printf("Yankees");
   }

}
```

// is for a comment

One class in the program must contain a main method.

Starting point for Java programs is the main method.

\n means go to the next line (can go anywhere in the string).

Like C++ “cout”

- Now we will move on to the sequence of creating and running a Java application…

Steps To Create and Run Java App

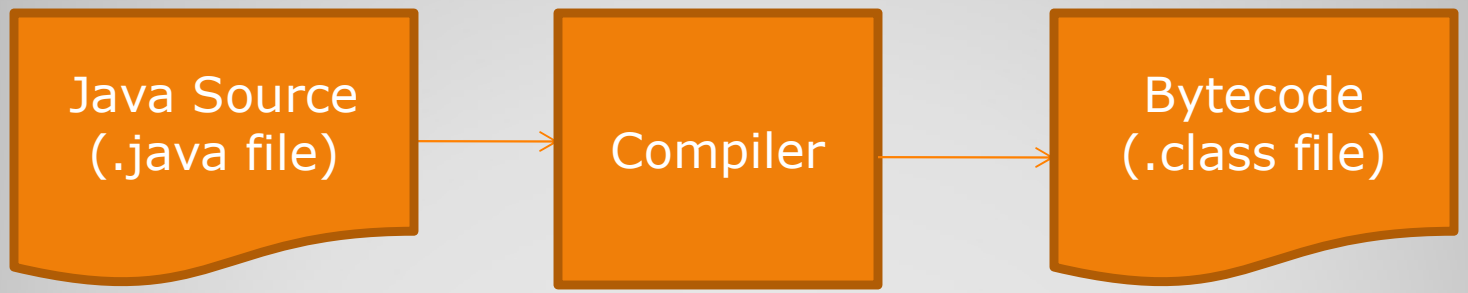
1. Type program in editor (write Java code)
2. Compile program (creates Java bytecode)
3. Load program (bytecode) into memory
4. Verify bytecode
5. Execute bytecode (actually run the app)

Steps To Create and Run Java App

- Write Java code in a text document
- Filename should have a ".java" extension

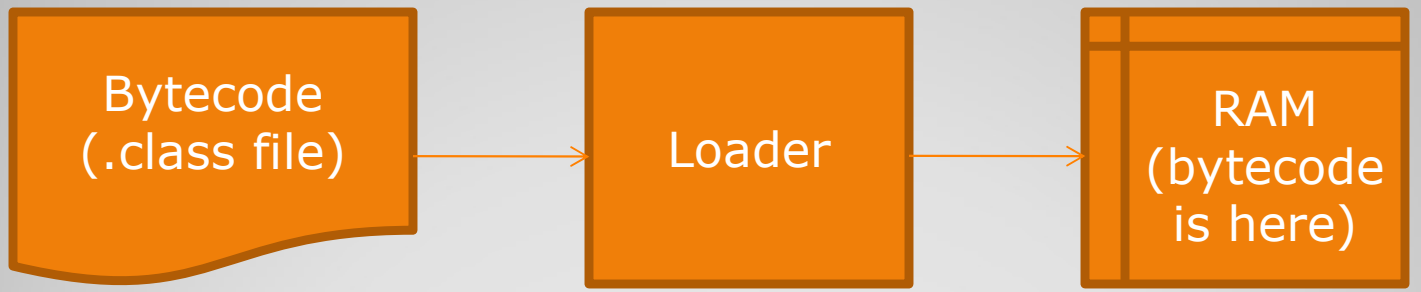
Type Program In Editor

- Compile the ".java" file (Java source)
- Translates Java code to bytecodes
- Bytecodes are lower level language (one step closer to machine language)
- Creates a ".class" file (bytecode)



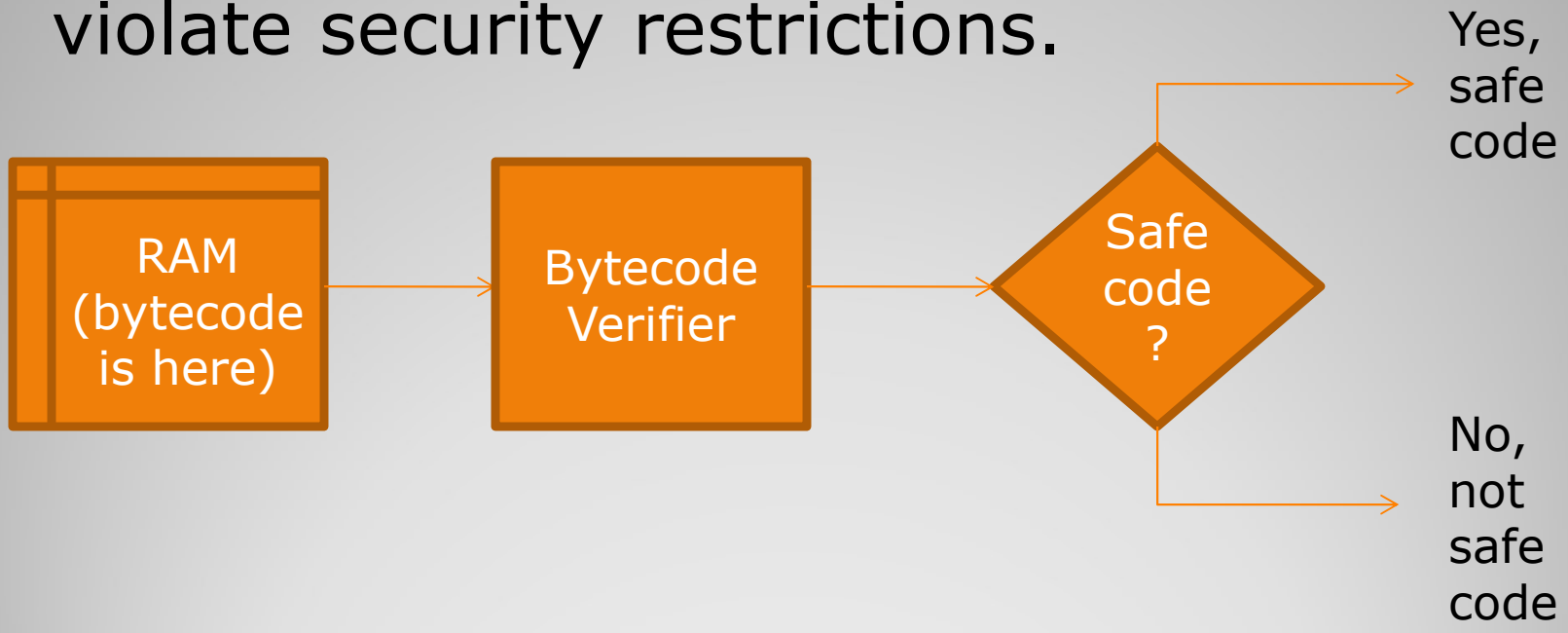
Compile

- Takes ".class" file (bytecode) as input
- Loader reads all the bytecode from the ".class" file and copies it into memory (RAM)



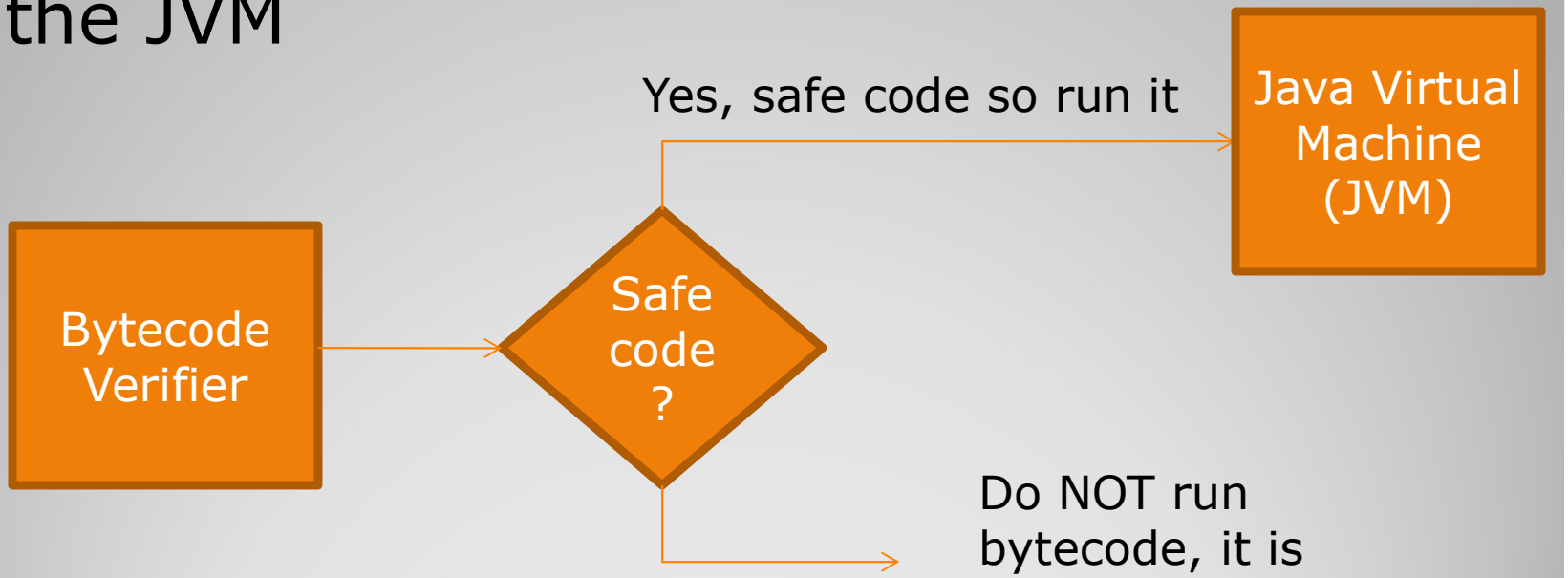
Loader

- Bytecode verifier examines the bytecode that is currently in memory (RAM)
- Makes sure that the bytecode does not violate security restrictions.

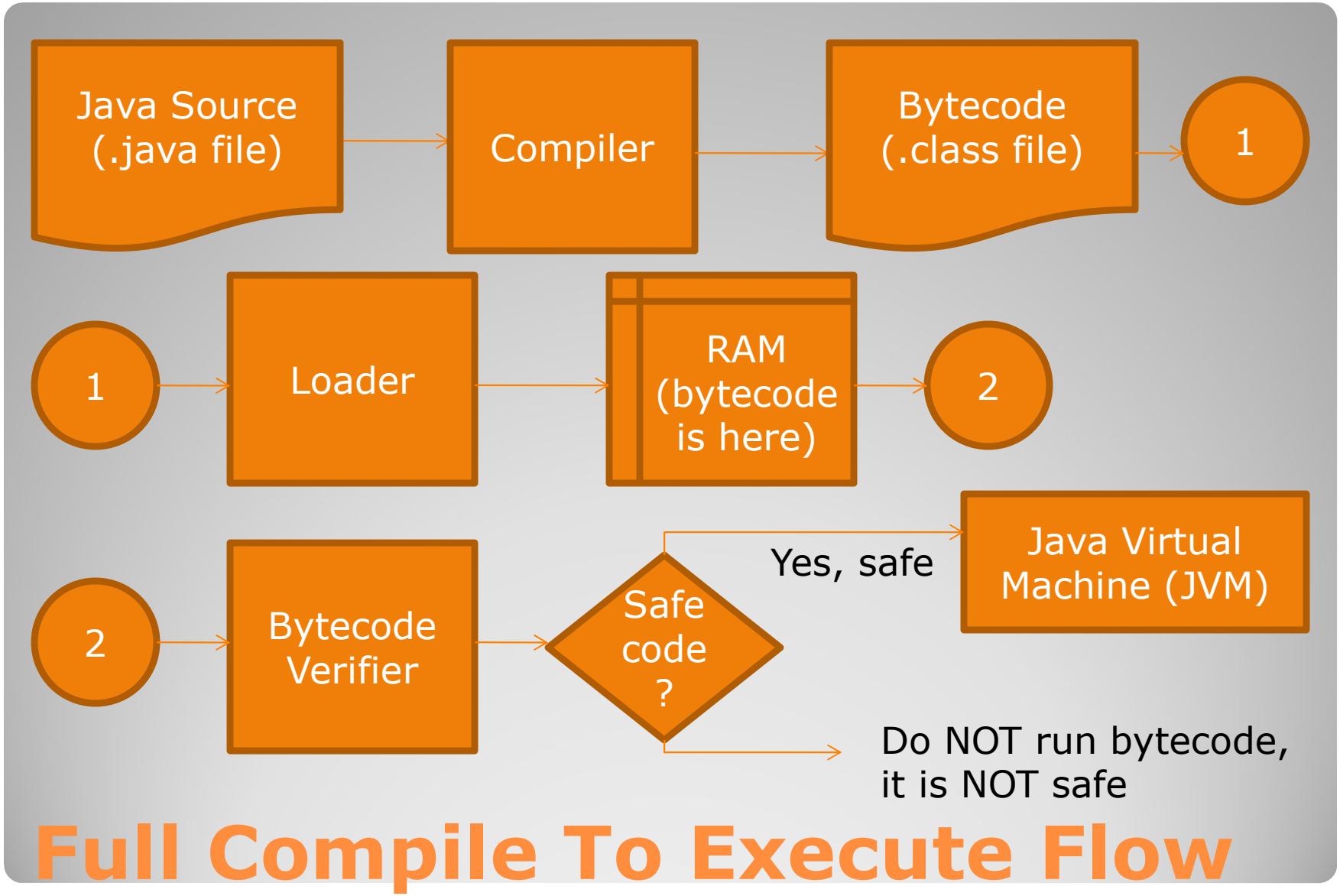


Bytecode Verification (Security)

- Java Virtual Machine (JVM) runs the bytecode
- The verified bytecode in RAM is input to the JVM



Execute (JVM)



Full Compile To Execute Flow

- End of Slides

End of Slides