Deen Dayal Upadhyaya College
Department of Computer Science
Workshop on Java J2SE

Dr. Rajni Bala
Associate Professor
Dept. of Computer Science
DDUC

JAVA

WRITE ONCE RUN ANYWHERE

Overview of Java

- Java one of the most widely used language for web application
- It is developed by James Ghosling, Patrick Naughton, Mike at Sun Microsystems in 1991.
- Initial name was Oak but was renamed to Java in 1985.
- Initially designed for use in Consumer electronics.
- Due to platform independence now used in web application

Applications of Java

- Developing Desktop application
- Web Applications like Linkedin
- Mobile OS like Andriod
- Embedded System
- Robotics and games etc.

Features of Java

- Simple
- Object-oriented
- Robust (compile-time and run-time checking, Memory management)
- Platform Independent
- Secure (enables us to develop virus free, temper free java programs)
- Multithreading
- Portable
- Architectural Neutral
- High performance (just-in time complier)

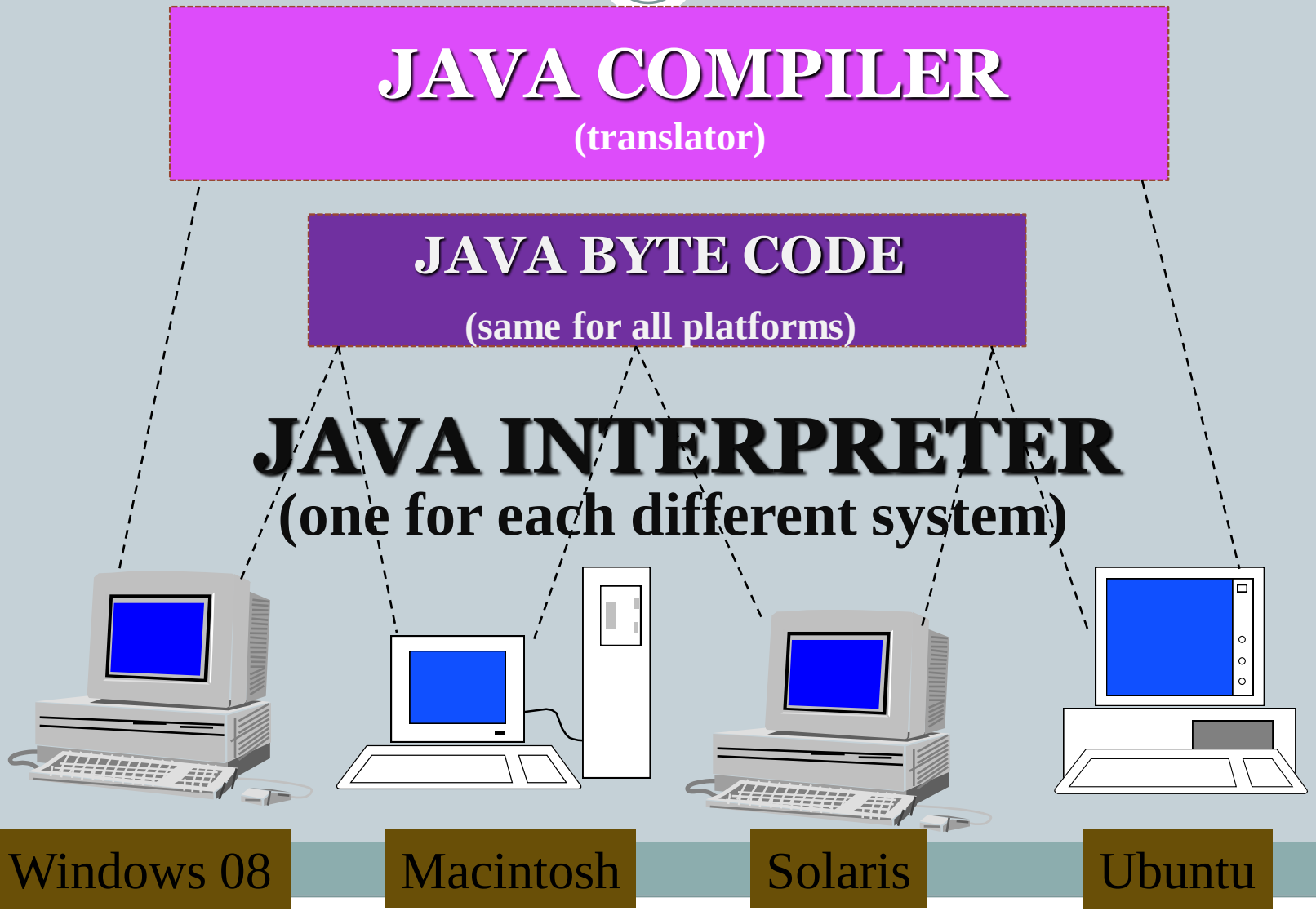
Java as Object Oriented

- “Objects all the way down”
- Simple and Familiar: “C++ Lite”
- No Pointers!
- Garbage Collector
- Dynamic Binding
- Single Inheritance with “Interfaces”

Java as Portable

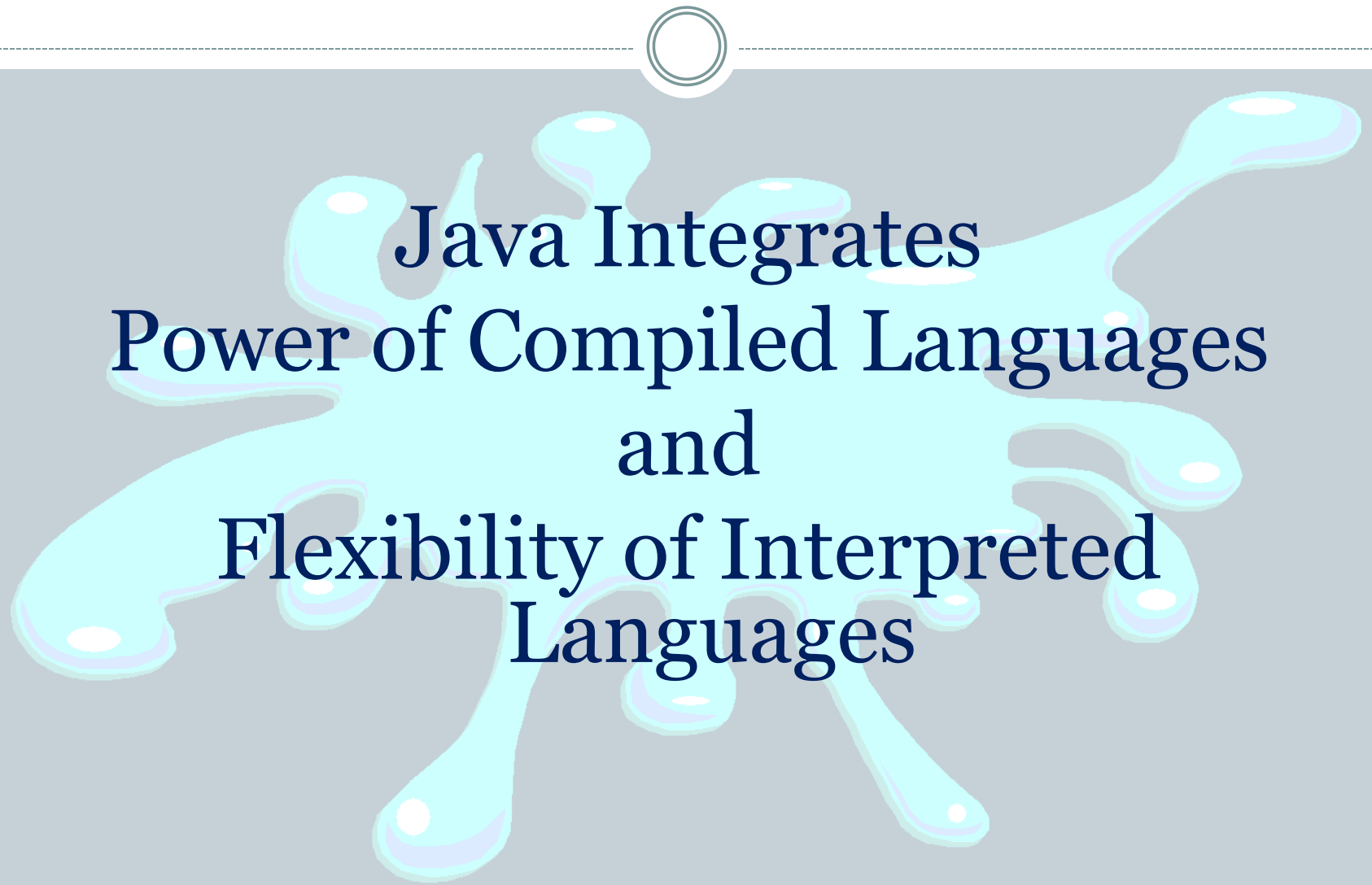
- Unlike other language compilers, Java complier generates code (byte codes) for Universal Machine.
- Java Virtual Machine (JVM): Interprets bytecodes at runtime
- Architecture Neutral
- No Link Phase
- Higher Level Portable Features: AWT, Unicode

Total Platform Independence



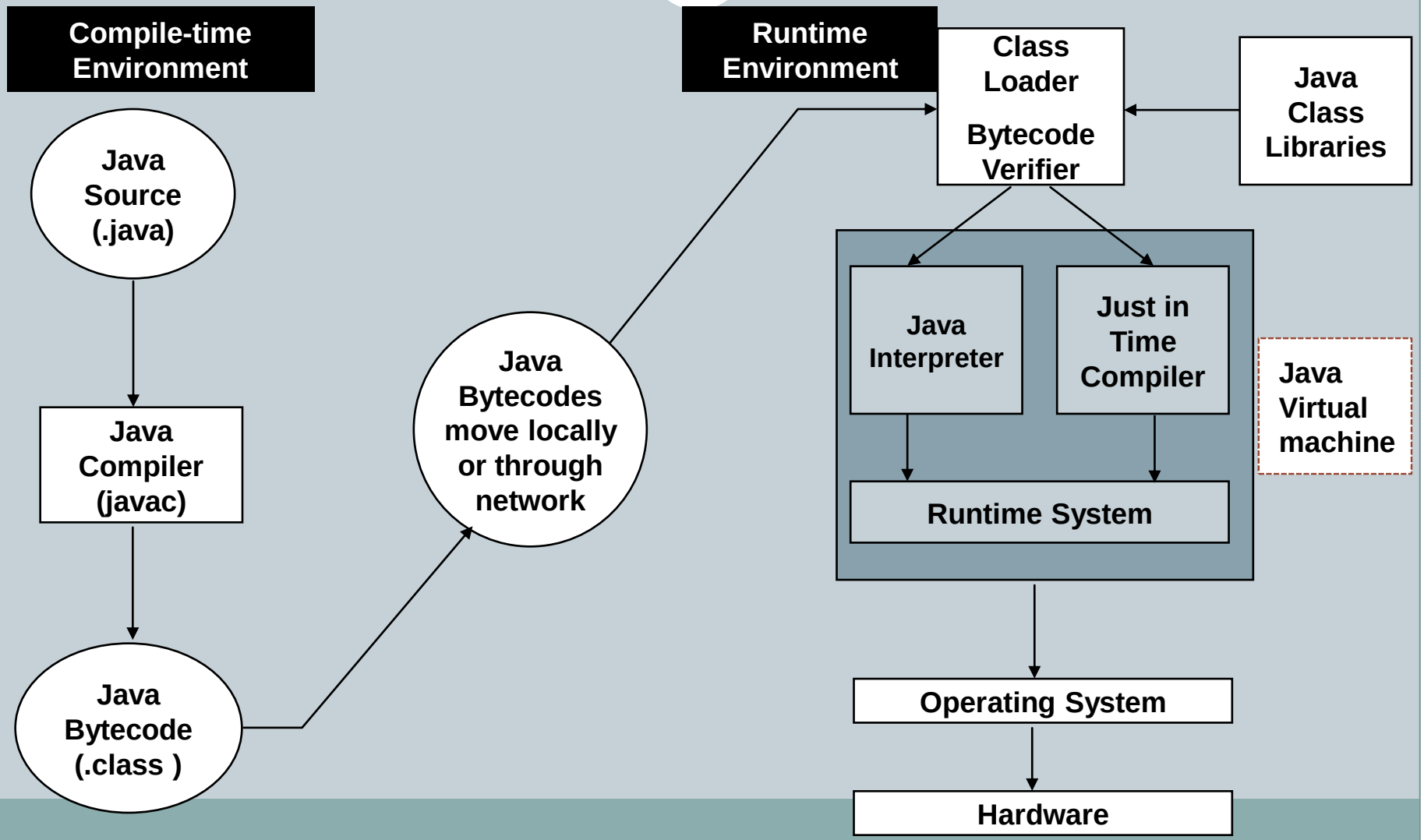
Java

Write Once, Run Anywhere.



Java Integrates Power of Compiled Languages and Flexibility of Interpreted Languages

Java Environment/ Life Cycle of Java Code



Architecture Neutral & Portable

- Java Compiler -Java source code to *bytecode*
- *Bytecode* - an intermediate form, closer to machine representation
- A virtual machine on any target platform interprets the bytecode
- Porting the java system to any new platform involves writing an interpreter that supports the Java Virtual Machine
- The interpreter will figure out what the equivalent machine dependent code to run

Bytecode Verifier

- The JVM verifies all bytecode before it is executed. Called when class is first loaded in runtime environment. This verification consists primarily of three types of checks:

1. Branches are always to valid locations.
2. Data is always initialized and references are always type-safe
3. Access to "private" or "package private" data and methods is rigidly controlled.

JIT

- The JIT compiler reads the bytecodes in many sections (or in full, rarely) and compiles them dynamically into machine language so the program can run faster.
- Java performs runtime checks on various sections of the code and this is the reason the entire code is not compiled at once. This can be done per-file, per-function or even on any arbitrary code fragment.
- The code can be compiled when it is about to be executed (hence the name "just-in-time"), and then cached and reused later without needing to be recompiled.

Java as Secure

- Language designed as safe
- Strict compiler
- Dynamic Runtime Loading (Verifier)
- Runtime Security Manager

Security Manager

- Prevents unauthorized disk read/writes
- Restricts network access
- Other access restrictions (native methods)
- Implementation is browser dependent

General Language Features

- C/C++ like syntax
- No pointers
- Objects all the way down
- Objects request services of other objects through *messages*
- Messages result in invocation of class *methods*

Java better than C++ ?

- No typedefs, Defines, or Preprocessor
- No Global Variables
- No Goto statements
- No Pointers
- No Multiple Inheritance
- No Operator Overloading
- No Automatic Coercions
- No Fragile Data Types

Removed From C++

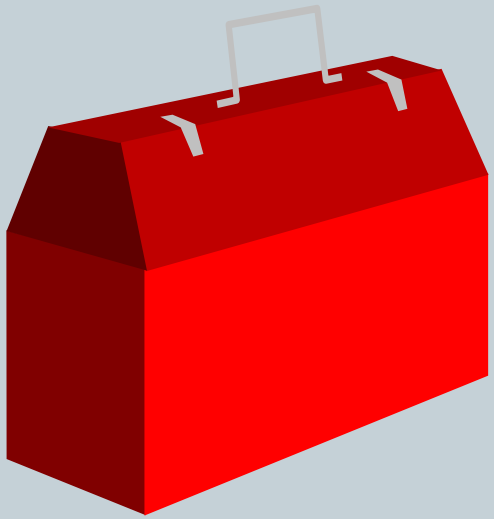
- Operator overloading
- Pointers and Array/pointers
- Multiple-inheritance of implementation
- Enum, typedef, #define
- Copy constructors, destructors
- Templates
- And other stuff….

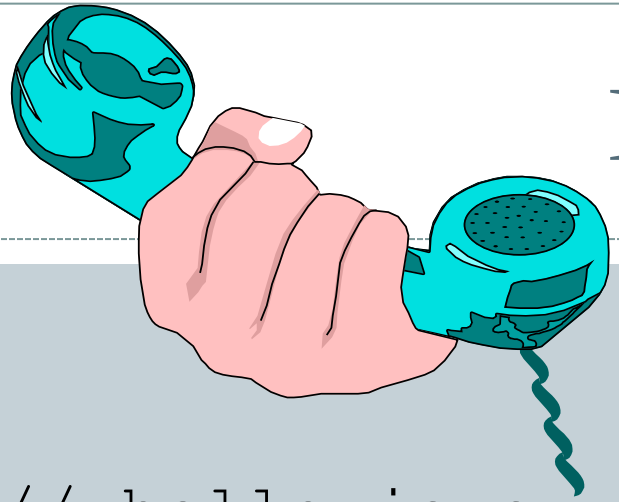
Added or Improved over C++

- Interfaces: type Vs. class
- Garbage collection
- Exceptions (More powerful than C++)
- Strings
- Instanceof
- Package
- Multi-threads

Java Development Kit

- javac - The Java Compiler
- java - The Java Interpreter
- jdb- The Java Debugger
- appletviewer -Tool to run the applets
- javap - to print the Java bytecodes
- javaprof - Java profiler
- javadoc - documentation generator
- javah - creates C header files





Hello Internet

```
// hello.java: Hello Internet program
class hello
{
   public static void main(String args[])
   {
      System.out.println("Hello Internet");
   }
}
```

Program Processing

- Save the file as hello.java
- Compilation

```
# javac hello.java
```

  results in hello.class

- Execution

```
# java hello
Hello Internet
#
```

Different Types of Java Programs

- Different ways to write/run a Java codes are:

Application- A stand-alone program that can be invoked from command line. A program that has a "main" method.

Applet- A program that resides on server and is executed on client. It contains no main. It can be executed either by applet viewer or by embedding in a web page.

Servlet- A program that resides on server and is also executed on the server. It is used for creating dynamic web-page. Usually used when interacting with databases.