# Intro to Java

ROUGH PAGES ARE NOT PART OF THE COURSE WORK

# Topics We Will Cover Today

History of Java

Why Java ?

Some Sample Java Based Applications

Writing a Basic Java Program

Java Program Development and Execution Steps

# History

Java

- Based on C and C++
- Developed in 1991 for intelligent consumer electronic devices
- Green Project (According to Gosling, "the goal was ... to build a system that would let us do a large, distributed, heterogeneous network of consumer **electronic devices all talking to each other**." )
- James Gosling Modified C++
- Named Oak then Java

# History….

Platform independent

Interpreted(تشريح) Language

Intermediate Code (Byte Code)

Reliable

Multiple inheritance and Operator overloading removed

No Pointers because of security reasons

Built in network support

# rough

## Bytecode vs Assembly Language

Bytecode is similar to assembly language in that it is not a high-level language, but it is still somewhat readable, unlike machine language. **Both may be considered "intermediate languages" that fall between source code and machine code**. The primary difference between the two is that bytecode is generated for a virtual machine (software), while assembly language is created for a CPU (hardware).

# History…

- Internet exploded(پھٹا) in 1993, saved project
  - Reliability
  - Security
  - Platform Independence
- Java formally announced in 1995
- Now used to create interactive web applications, distributed enterprise application for consumer devices (pagers, cell phones) and much more..

# Why Java???

# Motivation ☺

Portable WORA!!!!!! (**write once, run anywhere)**

Simple

“Pure” Object Oriented Language

Support for Distributed and Web Applications

Rich Libraries

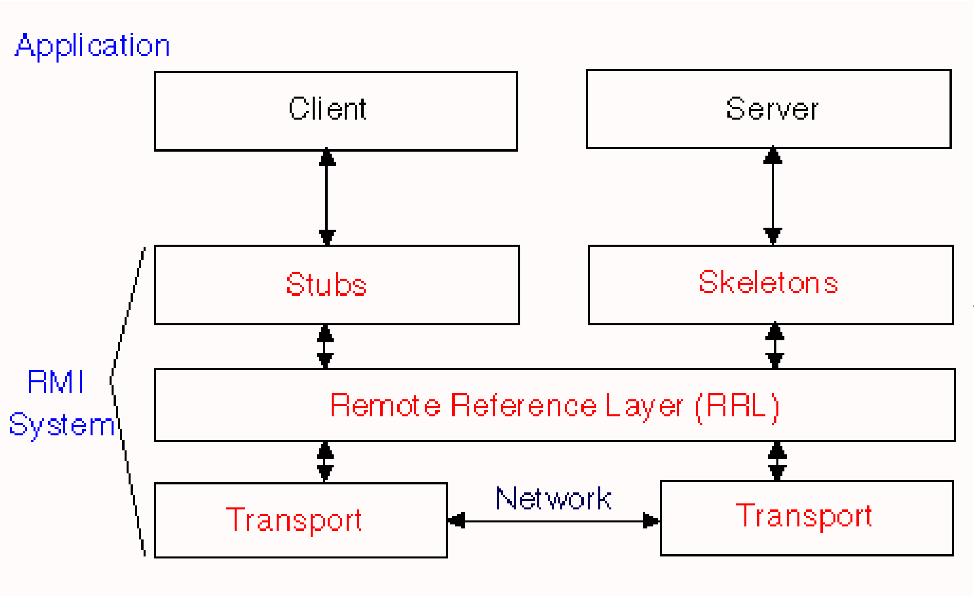
- Multithreading , Swing , RMI , NET , SQL …..

Automatic Garbage Collection

More Robust

# ROUGH

**RMI:**



**MultiThreading:** **MULTITHREADING in Java** is a process of executing two or more threads simultaneously to maximum utilization of CPU. **Multithreaded** applications execute two or more threads run concurrently.

# Portable

"Write-Once Run-Anywhere"

The Java Virtual Machine becomes the common denominator

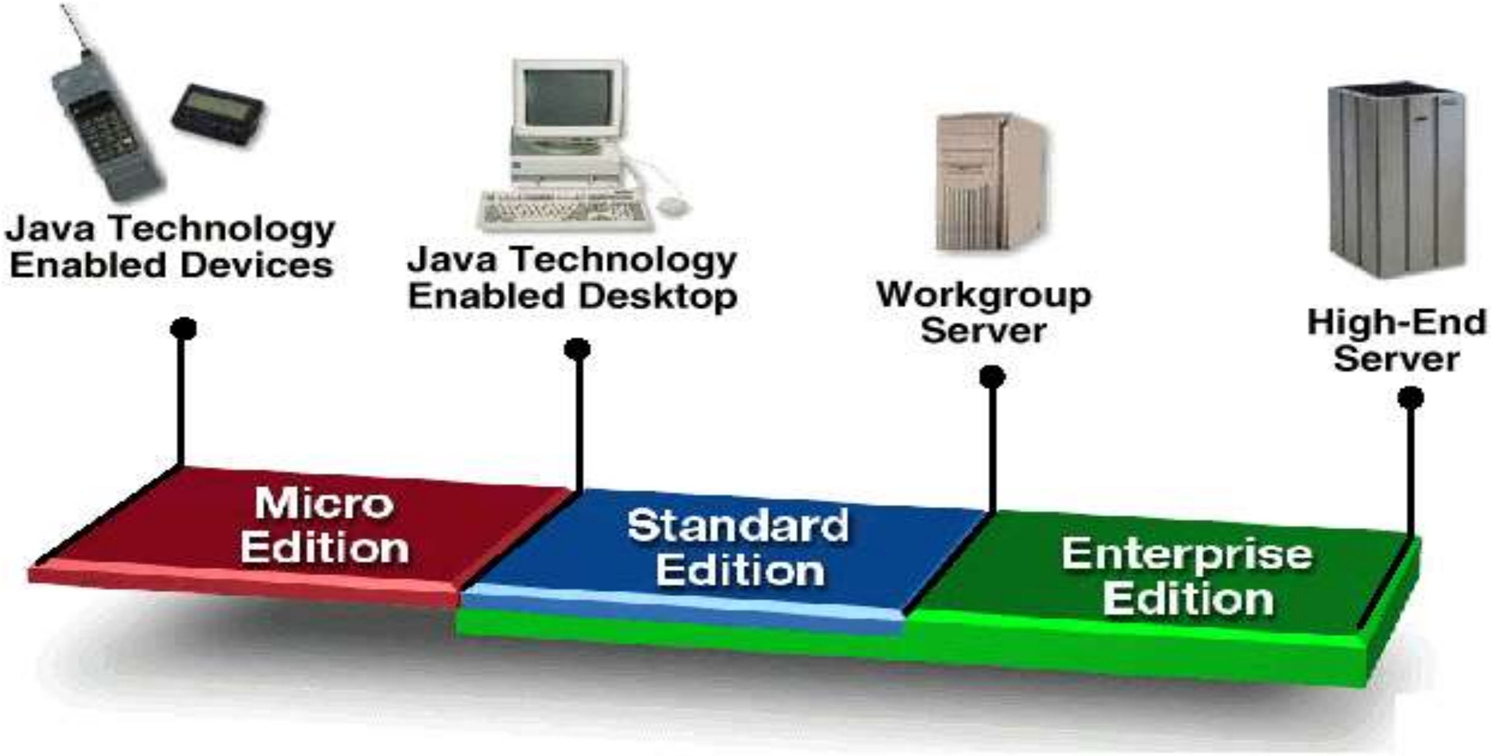
- Bytecodes are common across all platforms

  **Bytecode** is program code that has been compiled from source code into low-level code designed for a software interpreter.

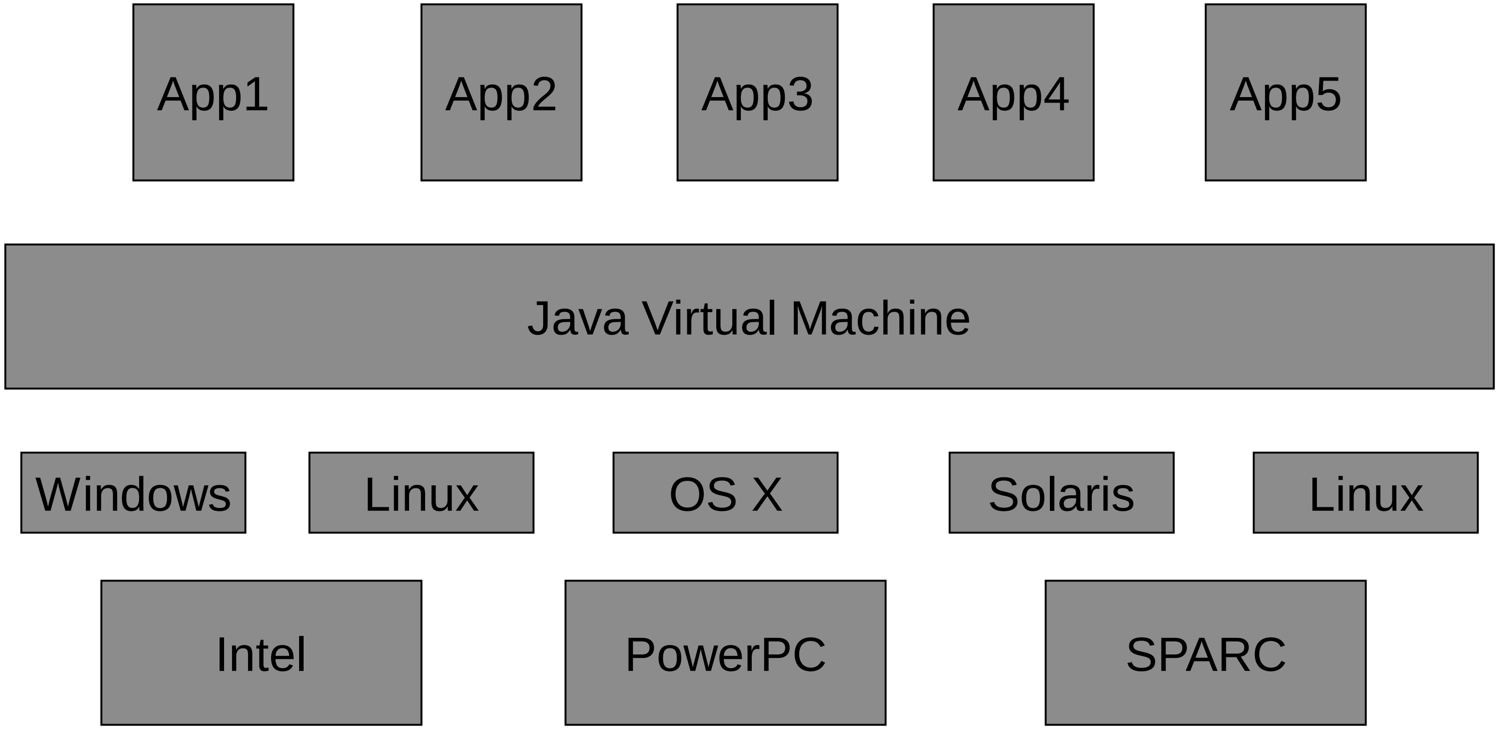
- Java Virtual Machine (JVM) hides the complexity of working on a particular platform
  - Difficult to implement a JVM
  - But simpler for the application developer

Java does this well

# Java Platform



# The Java Platform


App1

App2

App3

App4

App5

Java Virtual Machine

Windows

Linux

OS X

Solaris

Linux

Intel

PowerPC

SPARC

# Simple

Similar to C/C++ in syntax

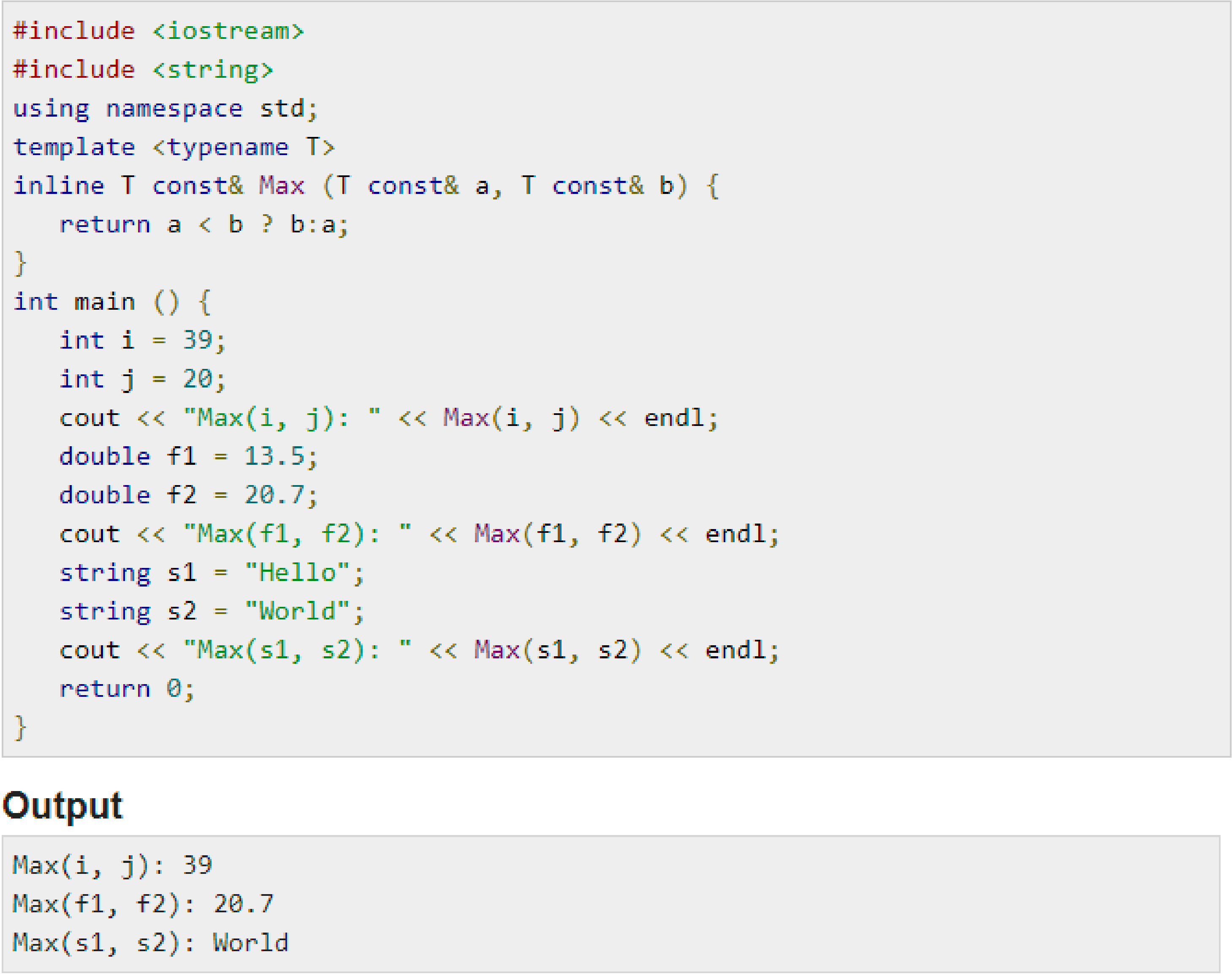
In-fact Java is C++ minus

- operator overloading
- direct pointer manipulation or pointer arithmetic
- multiple inheritance
- Destructors (Garbage Collector– handles memory automatically)
- No Templates
- Header/make files

Lots of other things which make Java more attractive.

# ROUGH

Template



```
#include <iostream>
#include <string>
using namespace std;
template <typename T>
inline T const& Max (T const& a, T const& b) {
   return a < b ? b:a;
}
int main () {
   int i = 39;
   int j = 20;
   cout << "Max(i, j): " << Max(i, j) << endl;
   double f1 = 13.5;
   double f2 = 20.7;
   cout << "Max(f1, f2): " << Max(f1, f2) << endl;
   string s1 = "Hello";
   string s2 = "World";
   cout << "Max(s1, s2): " << Max(s1, s2) << endl;
   return 0;
}
```

## Output

```
Max(i, j): 39
Max(f1, f2): 20.7
Max(s1, s2): World
```

# Object-Oriented

Fundamentally(بنیادی طور پر) based on OOP

All functions belong to classes or objects. No global variables or functions exist

All classes by default inherit from a common ancestor known as "Object"

"Almost" all data types are objects

OOP will be covered in a little more detail later.

# ROUGH

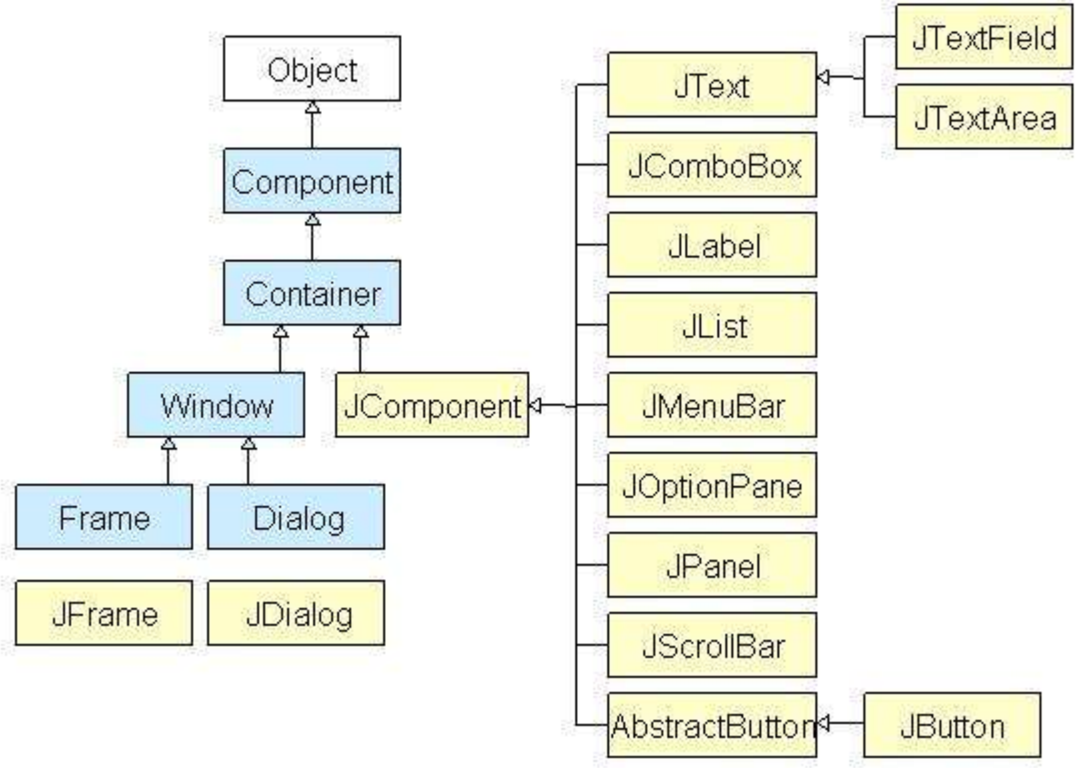
**Primitive Data Type:** such as boolean, char, int, short, byte, long, float, and double

**Non-Primitive Data Type or Object Data type:** such as String, Array, etc.

**Non**-**primitive** data types are not **defined** by the programming language, but are instead created by the programmer. They are sometimes called "reference variables," or "object references," since they reference a memory location, which stores the data.

# ROUGH

OBJECT



# Distributed / Network Oriented

## Java grew up in the days of the Internet

- Inherently network friendly
- Original release of Java came with Networking libraries
- Newer releases contain even more for handling distributed applications
  - RMI Remote Method Invocation **CHECK SLIDE 9**
  - , Transactions

# Support for Web and Enterprise Applications

Given below are some of the Java technologies that can be used for web and enterprise (for-profit business or company) application development

- Servlets
- JSP
- Applets
- JDBC
- RMI
- EJBs
- JSF

And many more…

# Robust / Secure / Safe

Designed with the intention of being secure

- *No pointer* arithmetic or memory management!
- The JVM “verifier”
  - Checks integrity(honesty) of byte-codes
- Dynamic(during run time) runtime checking for pointer and array access (non primitive data types **slide 16)**
  - No buffer overflow bugs! Since Java Strings are based on char arrays and Java automatically checks array bounds
- SecurityManager to check which operations a piece of code is allowed to do
- “Sandbox”(rules) operation for applets and other untrusted code
  - Limited set of operations or resources made available
  - Contrast(اس کے برعکس) to ActiveX

# ROUGH

A sandbox is an isolated testing environment that enables users to run programs or execute files without affecting the application, system or platform on which they run. Software developers use sandboxes to test new programming code. Cybersecurity professionals use sandboxes to test potentially malicious software.

**ActiveX** is a software framework from Microsoft (MSFT) that allows applications to share functionality and data with one another through web browsers, regardless of what programming language they're written in.

# Rich Set of Libraries

Multithreading

Swing

Regular Expression

NET

SQL

Util

Serialization …………….

# Java Programmer Efficiency

Faster Development

- More programmer friendly
- Less error prone

OOP

- Easier to manage large development projects

Robust(مضبوط) memory system

- No pointer arithmetic and manual memory management. Garbage collector!

Libraries

- Re-use of code

# Disadvantages ☹

Java performance IS slower than C/C++

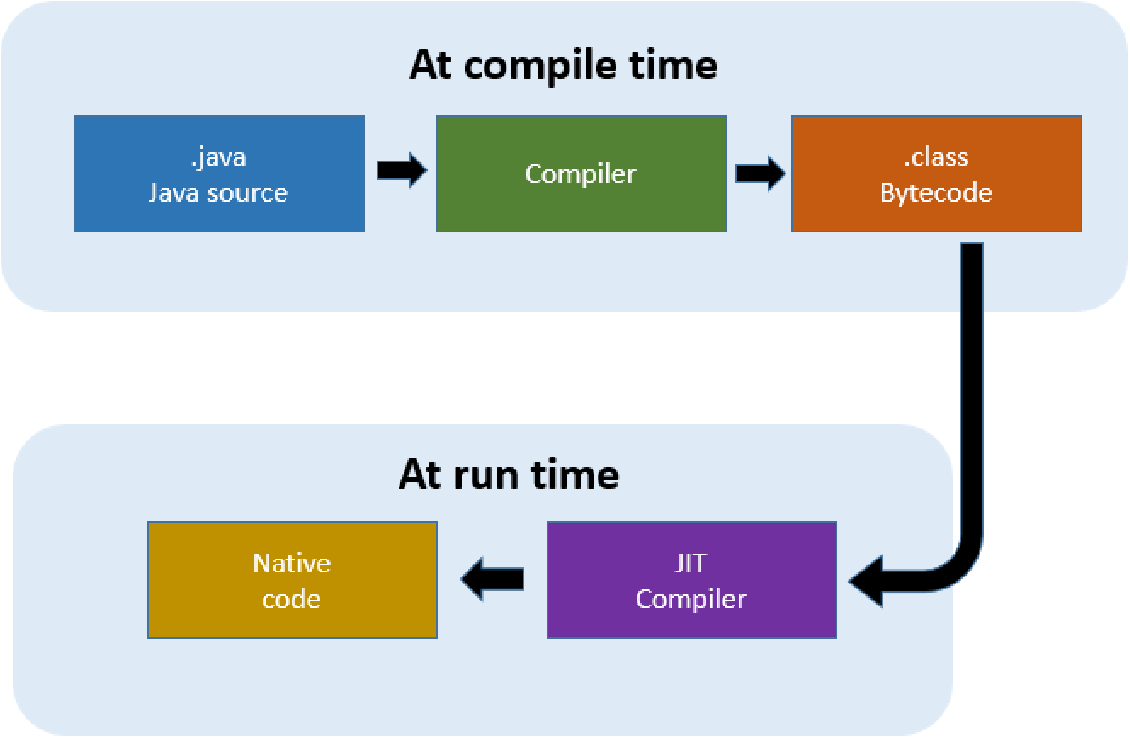
  - Tradeoff(تجارت) between development time vs. run time
  - Additional checks in Java which make is secure and robust and network aware etc, all have a small cost.

BUT

- JIT(Just In Time Compiler) compilation and HotSpot(by Oracle)
  - Dynamic compilation of bytecode to native code at runtime to improve performance
- HotSpot optimizes code on the fly based on dynamic execution patterns
  - Can sometimes be even faster than compiled C code!

Increasing processing speeds helps in overcoming this short fall

# ROUGH JIT Compiler



# Microsoft vs. Java

Java is platform independent

- Was considered a threat to Microsoft’s dominance
- Sun vs. Microsoft Law Suit

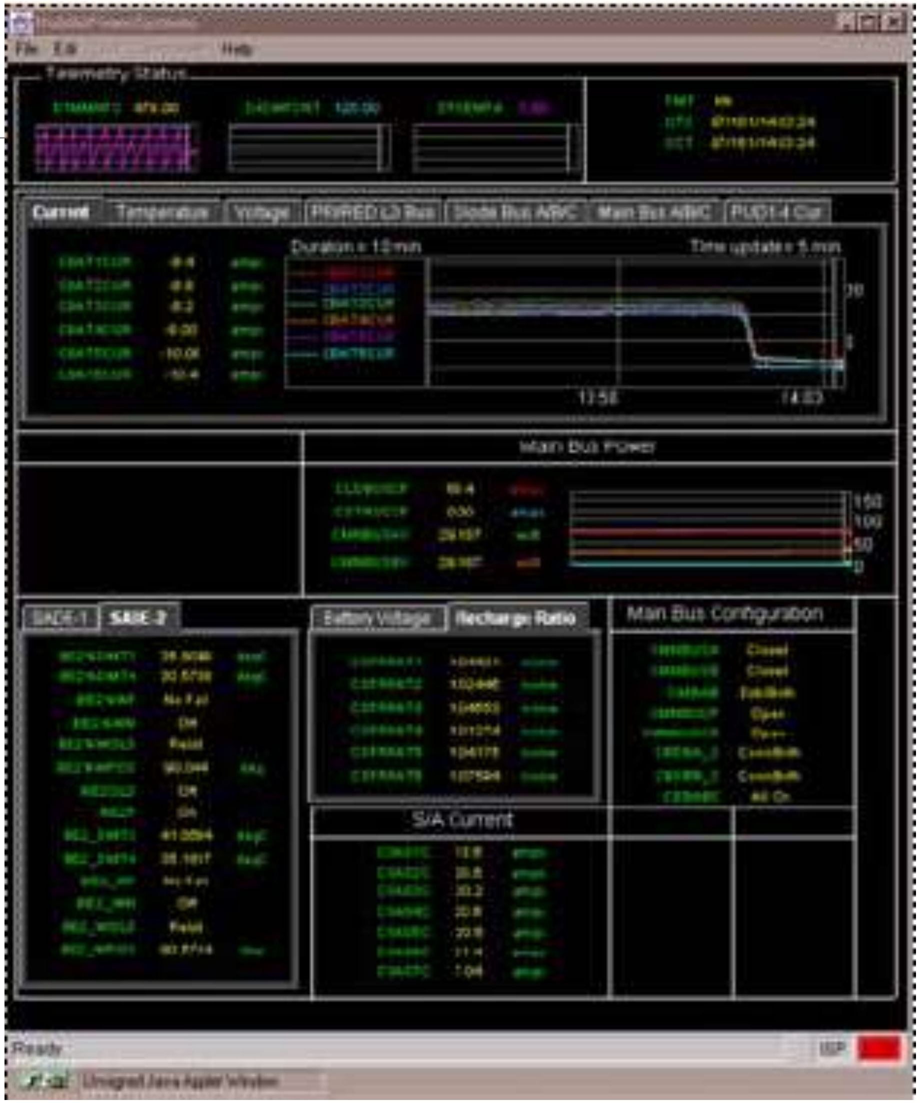
Microsoft’s latest response to Java

- C#
    - Very similar to Java in structure and style
    - Some improvements over past releases of Java (which have now emerged in Java 1.5)

# Some Sample Java Based Applications

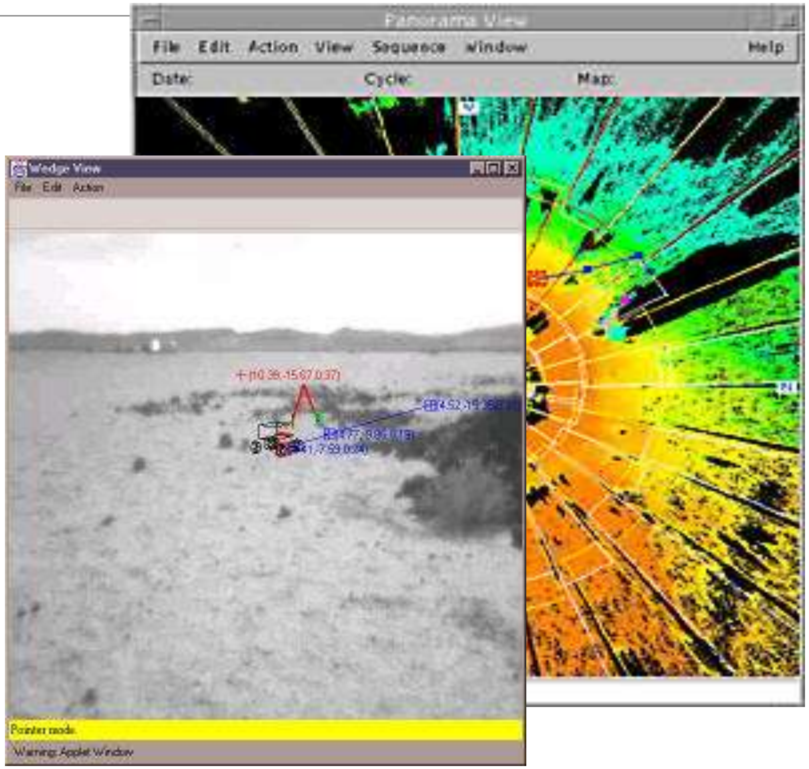
# Hubble Space Telescope Monitoring

- NASA Goddard’s most successful project ever
- Launched in 1990.
- Has sensitive light detectors and cameras
- Provided view of galaxies up to 10 billion light years away



# Mars Pathfinder Mission Simulator

- Used for world-wide data viewing
- Winner of the 1997 NASA software of the year
- The current rover location is displayed, along with visual indications of “targets”
- Provides close-ups of the wedge photograph

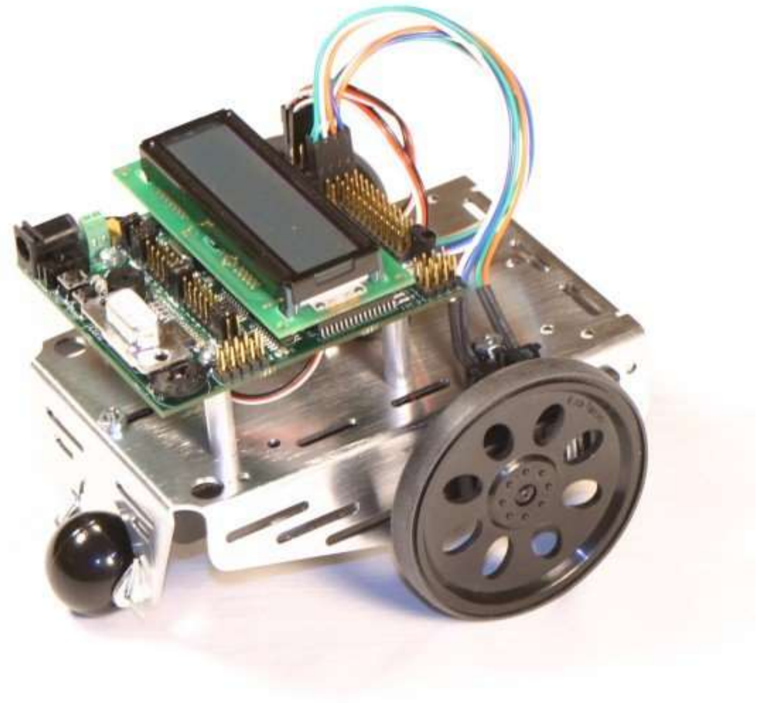


Pointer mode.

Warning: Applet Window

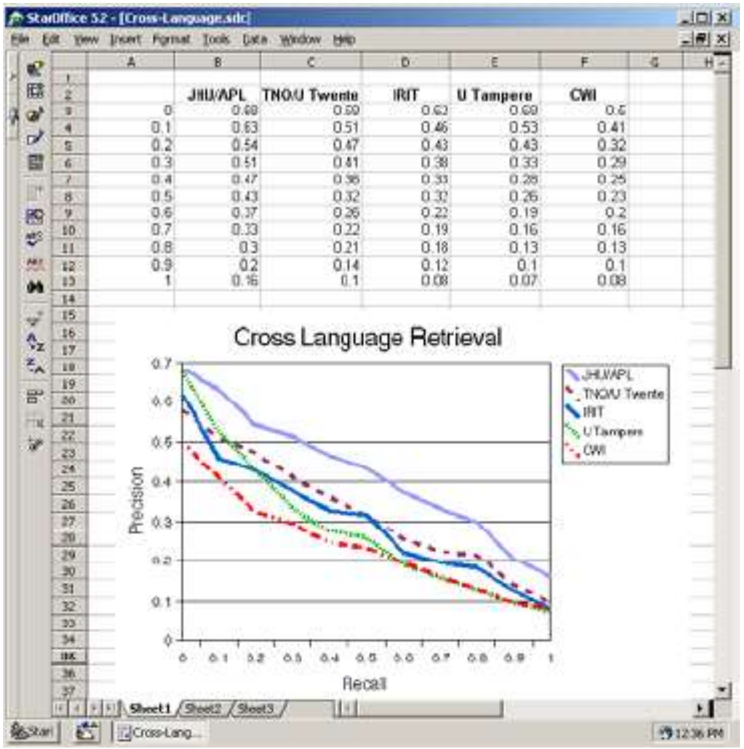
# IntelliBrain™ -Bot

- Java Programmable
- RoboJDE™ java enabled robotics software development environment
- Makes developing, debugging robotics program a snap(اچانک)



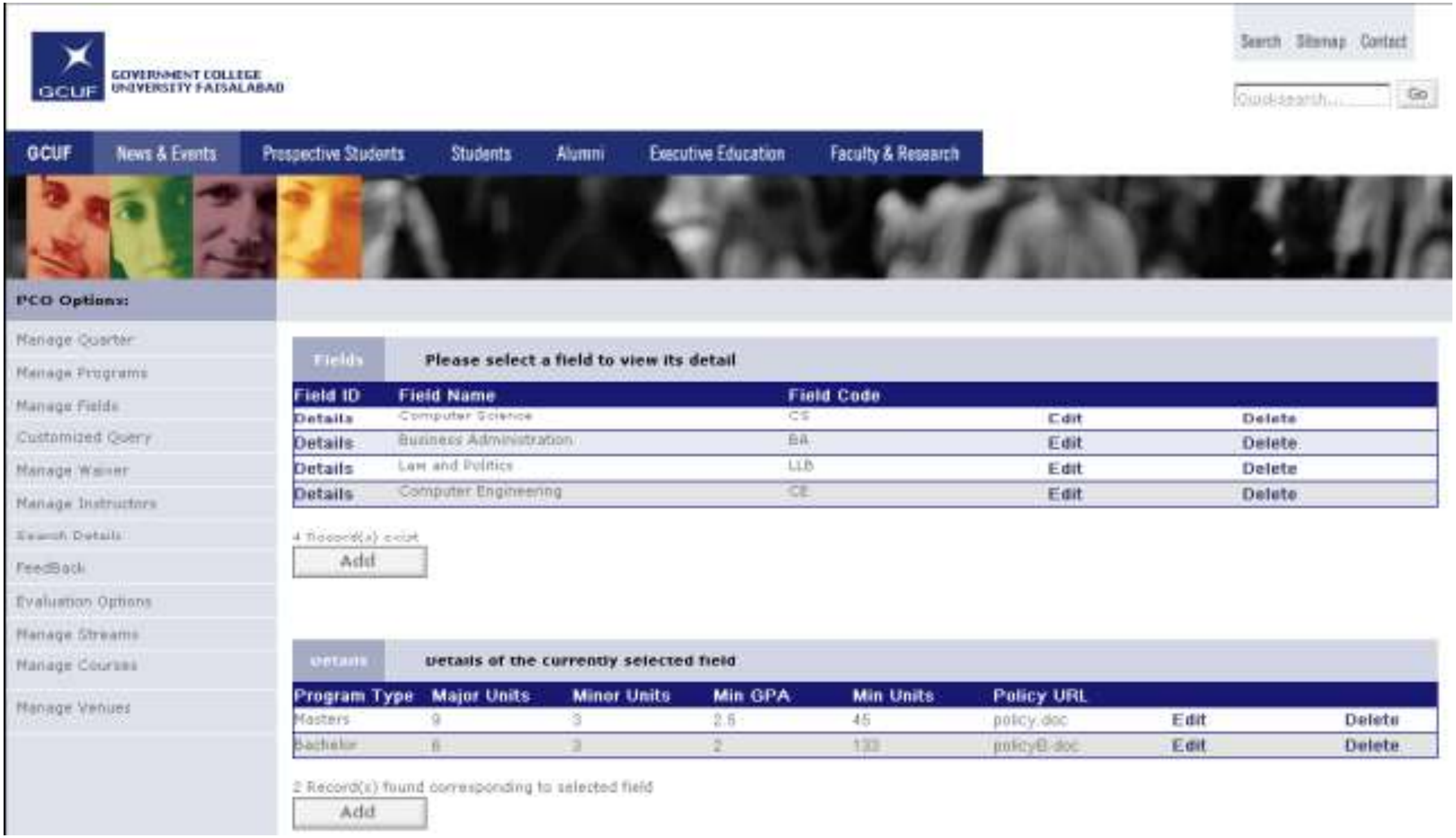
# Star Office 5.2

- Cross platform Office suite completely written in java



| | JHU/APL | TNO/U Twente | IRIT | U Tampere | CWI |
|---|---|---|---|---|---|
| 0 | 0.68 | 0.69 | 0.62 | 0.69 | 0.6 |
| 0.1 | 0.63 | 0.51 | 0.46 | 0.53 | 0.41 |
| 0.2 | 0.54 | 0.47 | 0.43 | 0.43 | 0.32 |
| 0.3 | 0.51 | 0.41 | 0.38 | 0.33 | 0.29 |
| 0.4 | 0.47 | 0.36 | 0.33 | 0.28 | 0.25 |
| 0.5 | 0.43 | 0.32 | 0.32 | 0.26 | 0.23 |
| 0.6 | 0.37 | 0.26 | 0.22 | 0.19 | 0.2 |
| 0.7 | 0.33 | 0.22 | 0.19 | 0.16 | 0.16 |
| 0.8 | 0.3 | 0.21 | 0.18 | 0.13 | 0.13 |
| 0.9 | 0.2 | 0.14 | 0.12 | 0.1 | 0.1 |
| 1 | 0.16 | 0.1 | 0.08 | 0.07 | 0.08 |

# Web Based
# Course Registration System



Fields Please select a field to view its detail

| Field ID | Field Name | Field Code | | |
|---|---|---|---|---|
| Details | Computer Science | CS | Edit | Delete |
| Details | Business Administration | BA | Edit | Delete |
| Details | Law and Politics | LLB | Edit | Delete |
| Details | Computer Engineering | CE | Edit | Delete |

4 Record(s) exist

Add

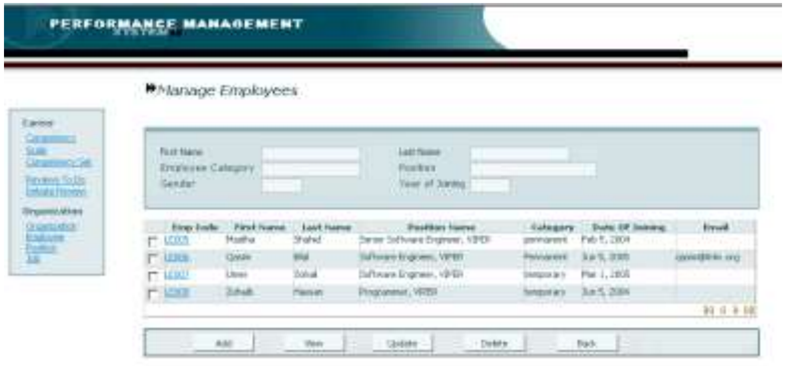
Details Details of the currently selected field

| Program Type | Major Units | Minor Units | Min GPA | Min Units | Policy URL | | |
|---|---|---|---|---|---|---|---|
| Masters | 9 | 3 | 2.5 | 45 | policy.doc | Edit | Delete |
| Bachelor | 6 | 3 | 2 | 133 | policyB.doc | Edit | Delete |

2 Record(s) found corresponding to selected field

Add

# Web Based Performance Review Management System

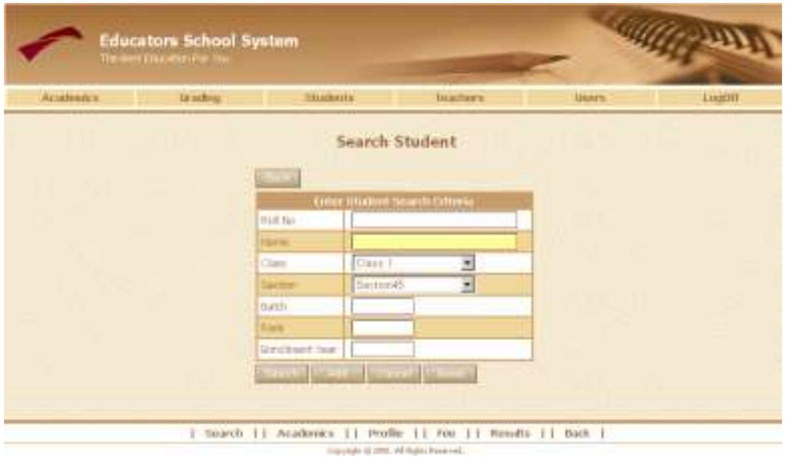


Manage Employees

| | | | |
|---|---|---|---|
| First Name | | Last Name | |
| Employee Category | | Position | |
| Gender | | Year of Joining | |

| | Emp Code | First Name | Last Name | Position Name | Category | Date Of Joining | Email |
|---|---|---|---|---|---|---|---|
| ☐ | LC005 | Madiha | Shahid | Senior Software Engineer, VPER | permanent | Feb 5, 2004 | |
| ☐ | LC006 | Qasim | Bilal | Software Engineer, VPER | Permanent | Jun 5, 2005 | qasim@[illegible].org |
| ☐ | LC007 | Umer | Sohail | Software Engineer, VPER | temporary | Mar 1, 2005 | |
| ☐ | LC008 | Zohaib | Hassan | Programmer, VPER | temporary | Jun 5, 2004 | |

# Web Based
# School Management System


Educators School System
The Best Education For You
Academics
Grading
Students
Teachers
Users
LogOff
Back
Search
Add
Cancel
Reset
| Search || Academics || Profile || Fee || Results || Back |
Copyright © 2005. All Rights Reserved.

Search Student

| Enter Student Search Criteria | |
|---|---|
| Roll No | |
| Name | |
| Class | Class I |
| Section | Section45 |
| Batch | |
| Rank | |
| Enrollment Year | |

# Writing Basic Java Program

## SYNTAX FOR C++  PROGRAMMERS

# Canonical Example HelloWorld Application in Java

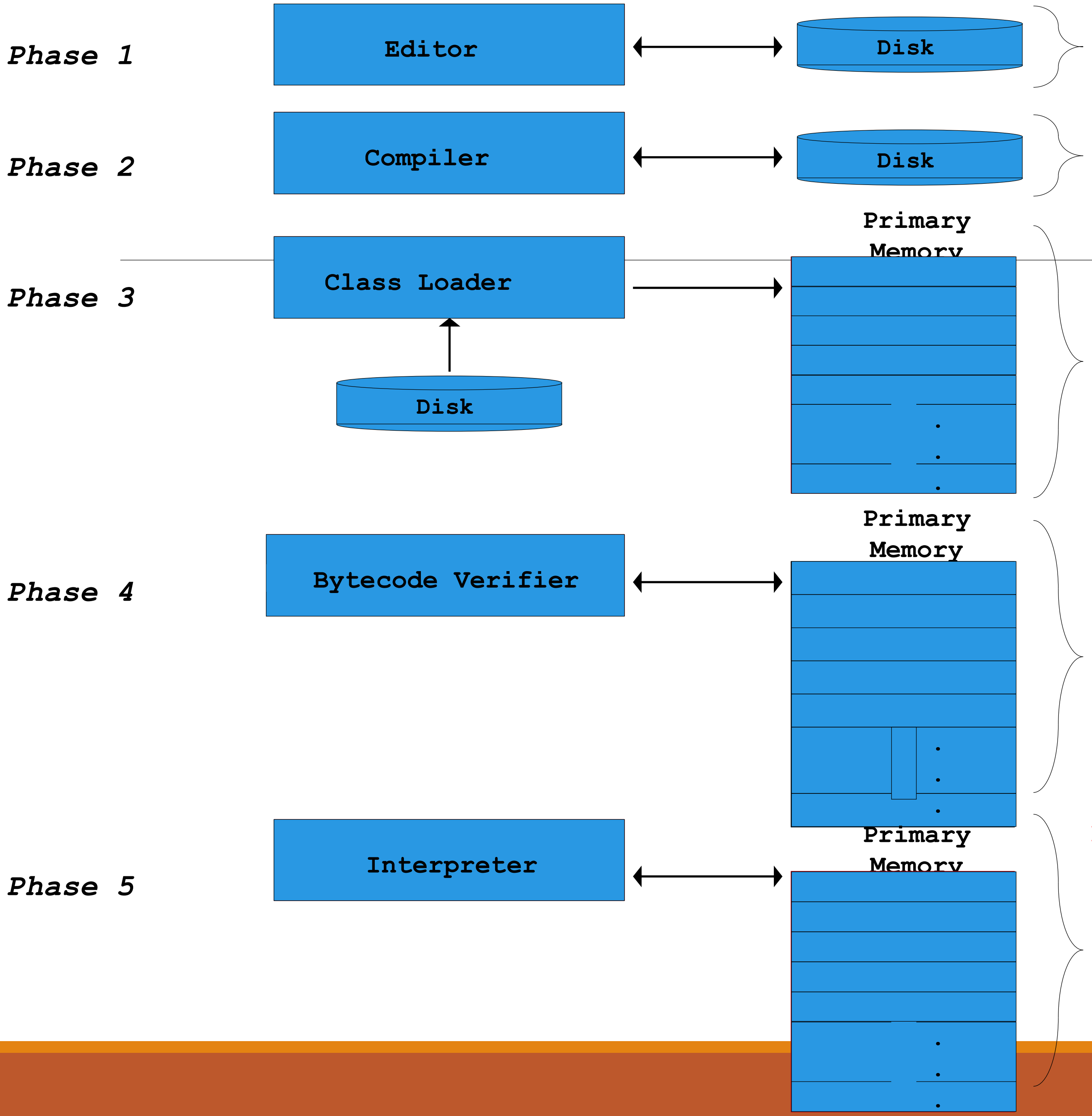
```
/* The HelloWorldApp class implements an application that
  simply displays "Hello World!" to the standard output. */

public class HelloWorldApp {

  public static void main(String[] args) {

                    //Display the string. No global main

        System.out.println("Hello World!");        }
}
```



Program is created in the editor and stored on disk.

Compiler creates bytecodes and stores them on disk.

Class loader puts bytecodes in memory.

Bytecode verifier confirms that all bytecodes are valid and do not violate Java's security restrictions.

Interpreter reads bytecodes and translates them into a language that the computer can understand, possibly storing data values as the program executes.