

Hello, Java

finalDesk

# Java Is

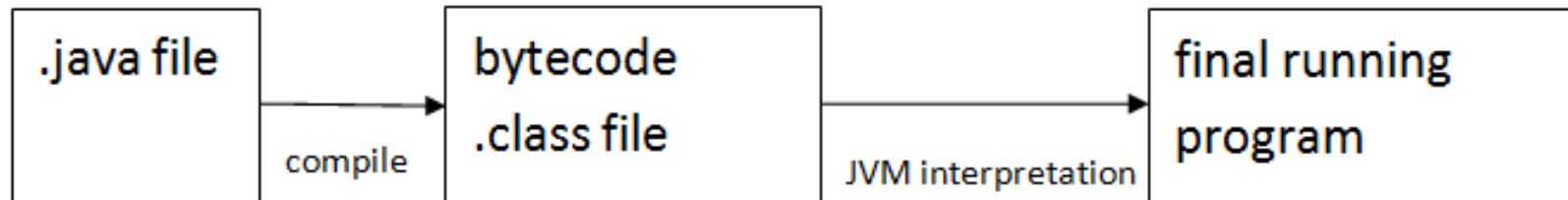
- A general purpose programming Language
  - many different application areas
- An execution platform
  - Java Virtual Machine
  - brings transparent portability to Java applications
- A library of software
  - standard library of components
  - huge amount of 3<sup>rd</sup> party supplied components
- An ecosystem
  - tools
  - community projects and activities

# The Java Programming Language

- Based on syntax and ideas of C/C++
- Many features not included
  - Direct access to memory
  - Multiple inheritance
  - Operator overloading
- Main language features
  - Largely object oriented
  - Exceptions
  - Multi-threading support
  - garbage collector for dynamic memory management
  - generic types
  - annotations for meta data

# The Java Virtual Machine

- Execution Platform for Java code
  - based on intermediate language – “bytecode”



# Java Libraries

- Much of Java's strength is in the libraries
  - Learning Java is really about learning its libraries
  - sometimes referred to as the Java APIs
- The libraries
  - Utilities for collections, including lists, queues, trees, etc
  - windowing tool-kits
  - Communication protocols
  - Web support
  - XML and web services
  - Many more
- Java code comprises packages of classes
  - classes define the “things” in the program
  - Packages organise classes into related groups

# Java Development Kit (JDK)

- Oracle provide the JDK
  - Includes tools and the standard class libraries
- Standard development tools

```
javac          //reference Java compiler
java           //reference Java
jar            //archive utility
javap          //class file disassembler
jdb            //Java debugger
javah          //create c headers & stub files
javadoc        //HTML document generator
appletviewer   //test environment for applets
```

# Integrated Development Environments

- IDEs for Java are widespread
  - Eclipse
  - NetBeans
  - IntelliJ IDEA
  - Jdeveloper
- Simplified application management
  - most IDEs hide details of directory structures
  - Integrated test and build tools
  - GUI building capabilities

# A First Java Program

```
class FirstProgram {  
    public static void main(String args[]) {  
        System.out.println("First Program");  
    }  
}
```

- The first line of the program contains keyword **class**.  
**FirstProgram** is an identifier that is the name of the class
- You can have more than one class per source file but only one can be public
- Name of the source file should be same as the name of the public class file



# public static void main(String args[])

- **public** keyword is an access specifier
- **static** allows main to be called without any instance
- **void** simply tells the compiler that main() does not return a value
- **String args[]** , which is an array of instances of the class String, **args** is used to receive any command line arguments

# Compiling and Running

- **javac** processes one or more Java source files
- **PATH** variable should be set before using javac
- **PATH** tells the windows system about the executables
- For Java it should be pointing to bin in jdk
- Classes should be available in classpath

```
javac FirstProgram.java
```

# Compiling and Running

- `javac` processes one or more Java source files
- Compiler outputs class files consisting bytecodes
- Execute the compiled program using the JVM
  - library code is loaded on demand

```
java FirstProgram
```

```
First Program
```

# Questions

- What is JAVA?
- What is JVM?
- What is API?
- What is bytecode?
- Is Java truly object oriented?
- Any IDE's you know for JAVA?

# Contact Info

- [trainers@finaldesk.com](mailto:trainers@finaldesk.com)
- [rishabh@finaldesk.com](mailto:rishabh@finaldesk.com)
- [nilesh@finaldesk.com](mailto:nilesh@finaldesk.com)
- [jignesh@finaldesk.com](mailto:jignesh@finaldesk.com)
- [yash@finaldesk.com](mailto:yash@finaldesk.com)
- [anand@finaldesk.com](mailto:anand@finaldesk.com)