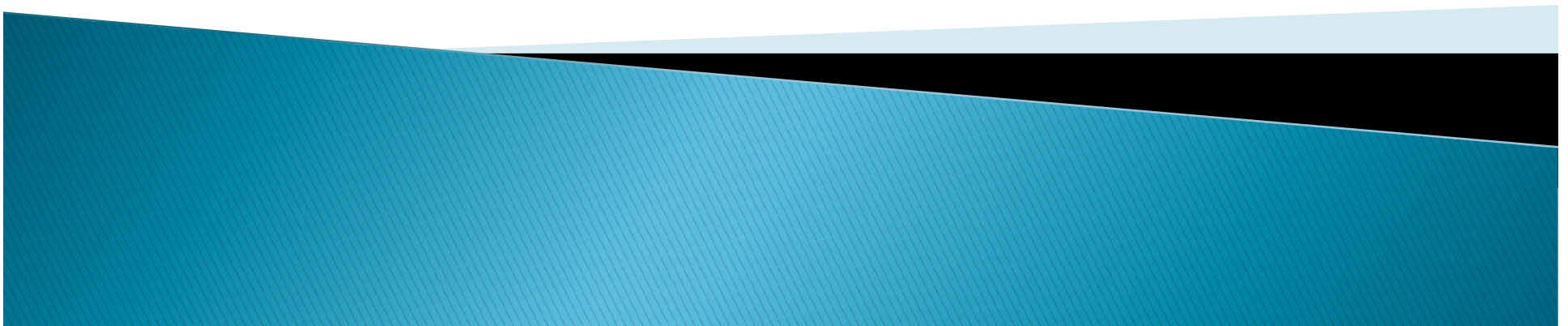


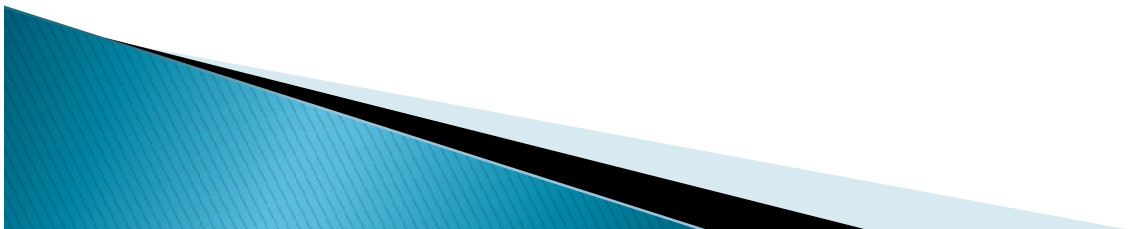
INTRODUCTION TO JAVA PROGRAMMING

Presented by
Smt Reetanjali Panda
Lecturer,UCPES



Overview of Java

- ▶ Java is one of the world's most important and widely used computer languages, and it has held this distinction for many years. Unlike some other computer languages whose influence has wearied with passage of time, while Java's has grown.
- ▶ As of 2020, Java is one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers using and working on it.



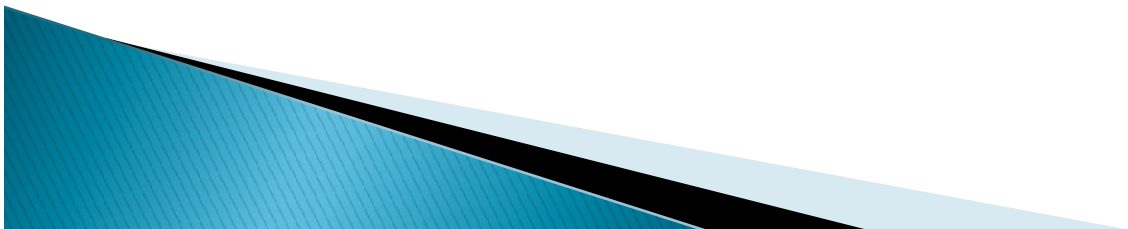
History of Java

- ▶ Java was developed by James Gosling, Patrick Naughton, Mike Sheridan at Sun Microsystems Inc. in 1991. It took 18 months to develop the first working version.
- ▶ The initial name was Oak but it was renamed to Java in 1995 as OAK was a registered trademark of another Tech company. Sun Microsystems Inc released the first public implementation as Java 1.0 in 1995. It is a Write Once, Run Anywhere(WORA) type language, providing no-cost run-times on popular platforms.



Evolution of Java

- ▶ Java was initially launched as Java 1.0 but soon after its initial release, Java 1.1 was launched. Java 1.1 redefined event handling, new library elements were added.
- ▶ In Java 1.2 Swing and Collection framework was added and suspend(), resume() and stop() methods were deprecated from Thread class.
- ▶ No major changes were made into Java 1.3 but the next release that was Java 1.4 contained several important changes. Keyword assert, chained exceptions and channel based I/O System was introduced.
- ▶ Java 1.5 was called J2SE 5, it added following major new features :
 - ▶ Generics
 - ▶ Annotations
 - ▶ Autoboxing and autounboxing
 - ▶ Enumerations
 - ▶ For-each Loop
 - ▶ Varargs
 - ▶ Static Import
 - ▶ Formatted I/O
 - ▶ Concurrency utilities



Evolution of Java(Cont.)

- ▶ Next major release was Java SE 7 which included many new changes, like :
 - ▶ Now **String** can be used to control Switch statement.
 - ▶ Multi Catch Exception
 - ▶ *try-with-resource* statement
 - ▶ Binary Integer Literals
 - ▶ *Underscore* in numeric literals, etc.
- ▶ And the latest addition to the lot is, **Java SE 8**, it was released on March 18, 2014. Some of the major new features introduced in JAVA 8 are,
 - ▶ Lambda Expressions
 - ▶ New Collection Package `java.util.stream` to provide Stream API.
 - ▶ Enhanced Security
 - ▶ Nashorn Javascript Engine included
 - ▶ Parallel Array Sorting
 - ▶ The JDBC-ODBC Bridge has been removed etc.



Application of Java

- ▶ Java is widely used in every corner of world and of human life. Java is not only used in softwares but is also widely used in designing hardware controlling software components. There are more than 930 million JRE downloads each year and 3 billion mobile phones run java.
- ▶ Following are some other usage of Java :
- ▶ Developing Desktop Applications
- ▶ Web Applications like [Linkedin.com](https://www.linkedin.com), [Snapdeal.com](https://www.snapdeal.com) etc
- ▶ Mobile Operating System like Android
- ▶ Embedded Systems
- ▶ Robotics and games etc.



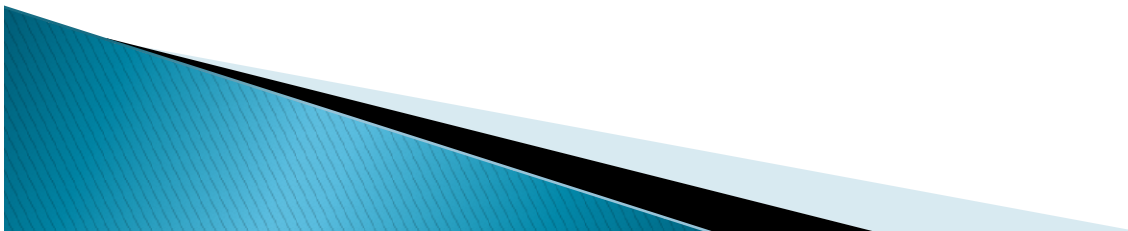
Features of Java

1) Simple

- ▶ Java is easy to learn and its syntax is quite simple, clean and easy to understand. The confusing and ambiguous concepts of C++ are either left out in Java or they have been re-implemented in a cleaner way.
- ▶ *Eg* : Pointers and Operator Overloading are not there in java but were an important part of C++.

2) Object Oriented

- Java is purely objected oriented language and implements all the object oriented principles. So in java maintenance of code is very easy. It helps the programmer to develop more reliable and error free code.



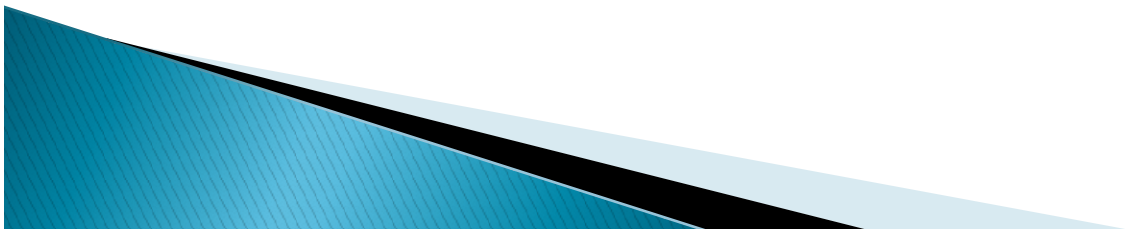
Features of Java(Cont.)

3) Robust

- ▶ Java makes an effort to eliminate error prone codes by emphasizing mainly on compile time error checking and runtime checking. But the main areas which Java improved were Memory Management and mishandled Exceptions by introducing automatic **Garbage Collector** and **Exception Handling**.

4) Architectural Neutral

- Compiler generates bytecodes, which have nothing to do with a particular computer architecture, hence a Java program is easy to interpret on any machine. Java compiler generates an architecture- neutral object file format which makes the compiled code to be executable on any processors, with the presence of Java runtime system.



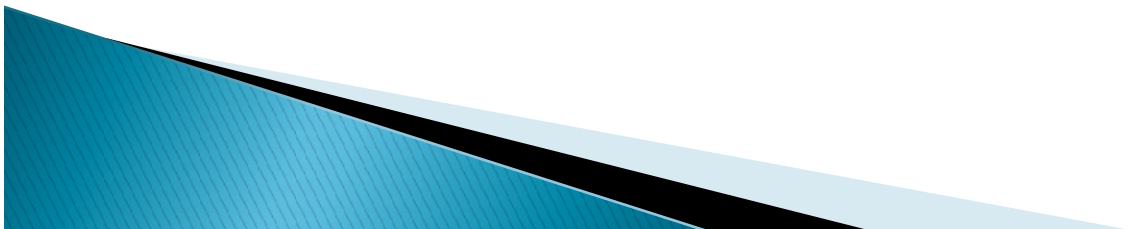
Features of Java(Cont.)

5) Portable

- ▶ Java Byte code can be carried to any platform. No implementation dependent features. Everything related to storage is predefined, example: size of primitive data types. Being architectural-neutral and having no implementation dependent aspects of the specification makes Java portable.

6) High Performance

- ▶ Java is an interpreted language, it enables high performance with the use of just-in-time compiler(JIT).



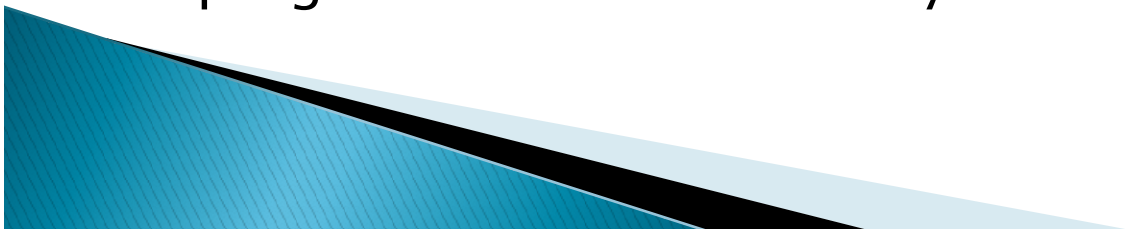
Features of Java(Cont.)

7) Secure

- ▶ When it comes to security, Java is always the first choice. With java secure features it enable us to develop virus free, temper free system. Java program always runs in Java runtime environment with almost null interaction with system OS, hence it is more secure.

8) Multi Threading

- ▶ Java multithreading feature makes it possible to write program that can do many tasks simultaneously. Benefit of multithreading is that it utilizes same memory and other resources to execute multiple threads at the same time, like While typing, grammatical errors are checked along. With Java's multithreaded feature it is possible to write programs that can do many tasks simultaneously.



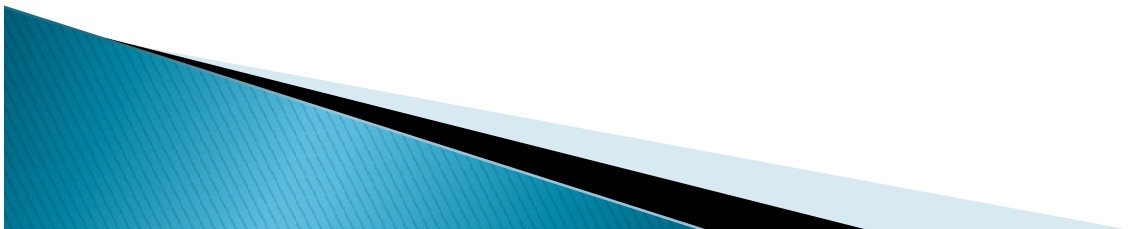
Features of Java(Cont.)

9) Platform Independent

- ▶ Unlike other programming languages such as C, C++ etc which are compiled into platform specific machines. Java is guaranteed to be write-once, run-anywhere language.
- ▶ On compilation Java program is compiled into bytecode. This bytecode is platform independent and can be run on any machine, plus this bytecode format also provide security. Any machine with Java Runtime Environment can run Java Programs.
- java is a Platform independent which is portable & easily executed on all operating systems (windows, Linux, solaris) or Java programs can run on any platform.

10) WORA(Write Once Run Anywhere)

- ▶ Java programs written once can be run on different platforms without making changes to the java programs. Only Java interpreter is changed depending on the platform.



Features of Java(Cont.)

11) Built in Graphics

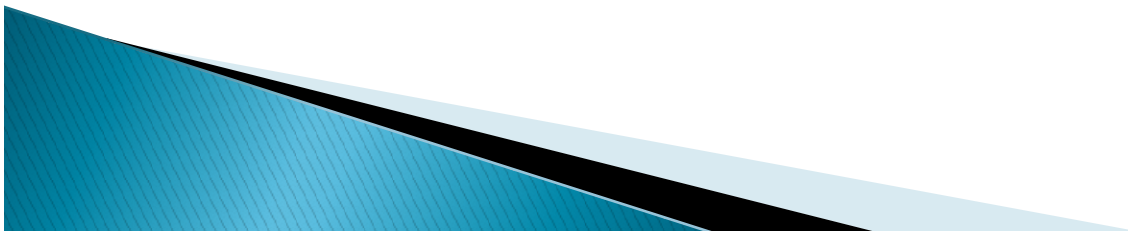
Java applications can use built in graphics features.

12) Light Weight Code: No huge coding is required.

13) Supports multimedia: Integration of audio, video, animation, graphics environment in java applications can be made available.

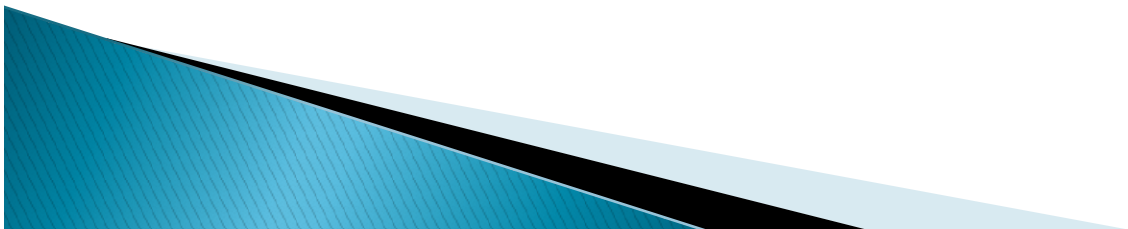
14) Open Product: Java is freely available in the internet.

15) Distributed: Information is distributed in various computers on a network because java is platform independent and can handle protocols like TCP/IP . Java is designed for the distributed environment of the internet.



Features of Java(Cont.)

- 16) **Dynamic:** Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment. Java programs can carry extensive amount of run-time information that can be used to verify and resolve accesses to objects on run-time.
- 17) **Interpreted:** Java byte code is translated on the fly to native machine instructions and is not stored anywhere. The development process is more rapid and analytical since the linking is an incremental and light weight process.

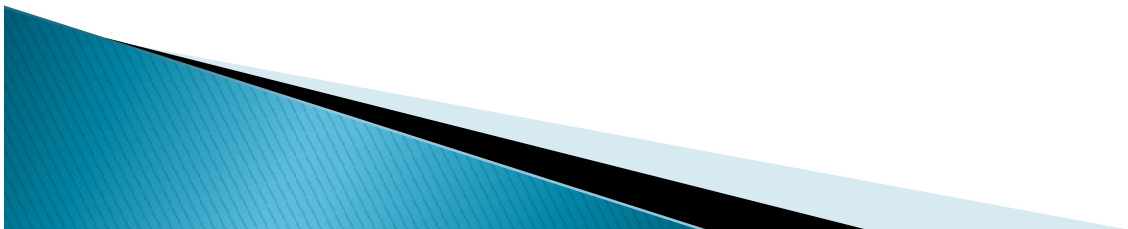


Running a Java Program

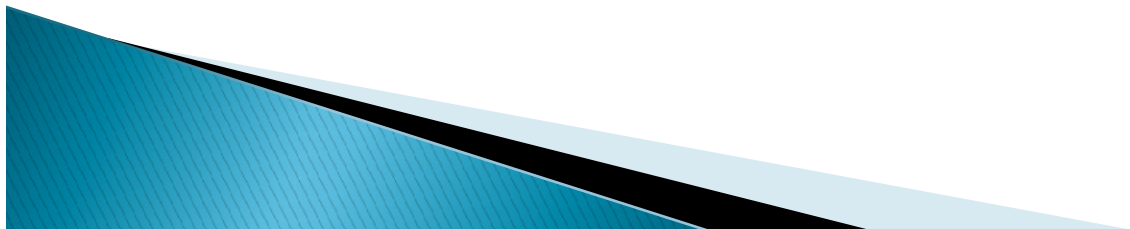
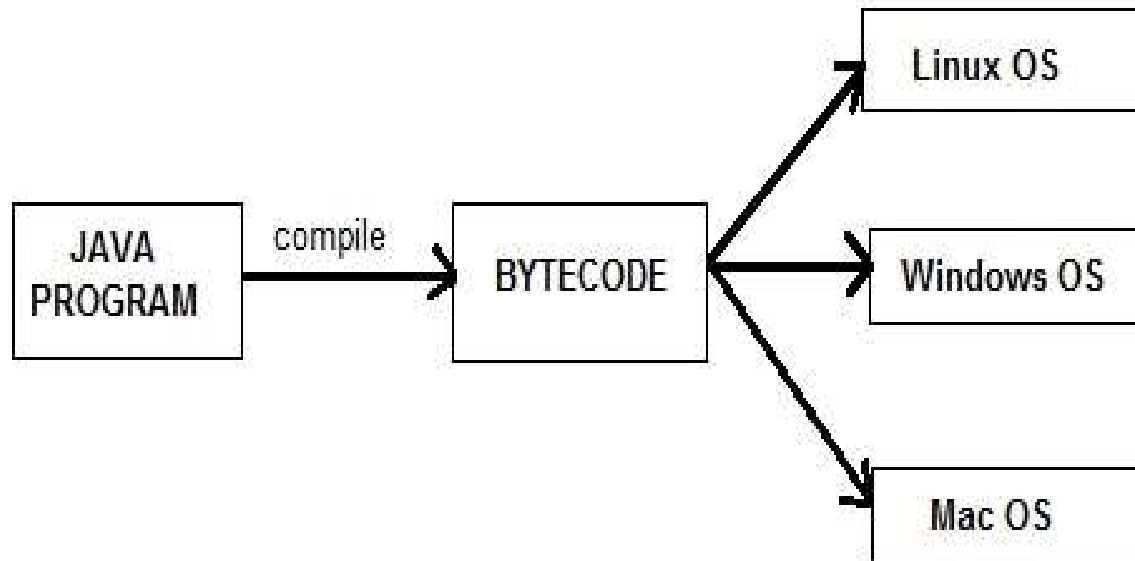
Unlike other programming languages, compiling Java source code does not result in a machine language program. Instead, when Java source code is compiled, we get what is called Java *bytecode*.

*Java bytecode is a form of machine language instructions. However, it is not primitive to the CPU. Java bytecode runs on a program that mimics itself as a real machine. This program is called the *Java Virtual Machine (JVM) or Java Run-time Environment (JRE)*.*

This architecture makes Java bytecode runs on any machines that have JVM, independent of the OSs and CPUs. This means the effort in writing Java source code for a certain program is spent once and the target program can run on any platforms. (E.g. Windows, MacOS, Unix, etc.)

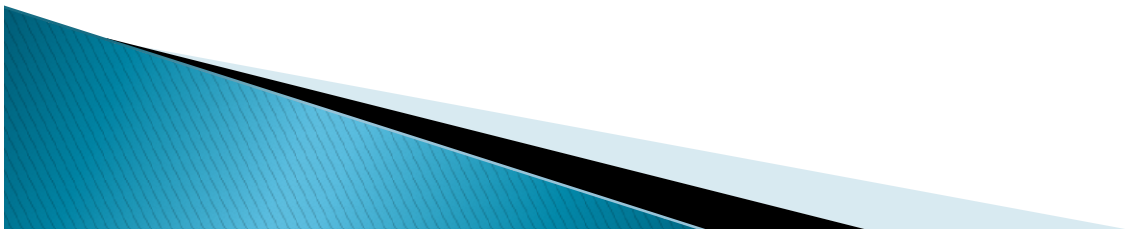


Running a Java Program(Cont.)



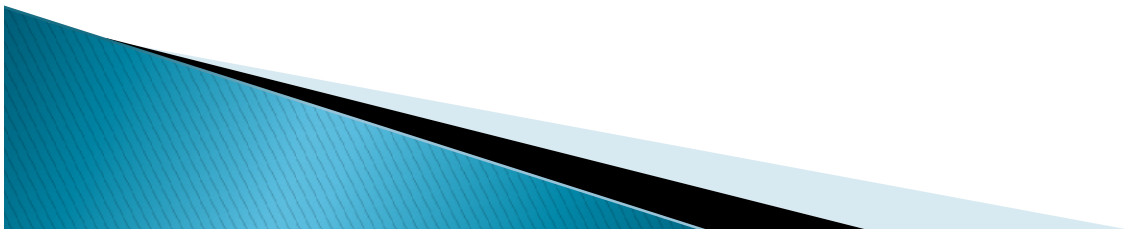
How is Java Platform Independent?

- ▶ Like C compiler, Java compiler does not produce native executable code for a particular machine. Instead, Java produces a unique format called bytecode. It executes according to the rules laid out in the virtual machine specification.
- ▶ Bytecode is understandable to any JVM installed on any OS. In short, the java source code can run on all operating systems that is code once compiled can run not only on all PC platforms but also mobiles or other electronic gadgets supporting java.



How Java Virtual Machine works?

- I. The source code written in java is saved as .java file.
- II. Using the java compiler the code is converted into an intermediate code called the byte code. The output is a .class file.
- III. This code is not understood by any platform, but only a virtual platform called the Java Virtual Machine.
- IV. This Virtual Machine resides in the RAM of your operating system. When the Virtual Machine is fed with this byte code, it identifies the platform it is working on and converts the byte code into the native machine code.



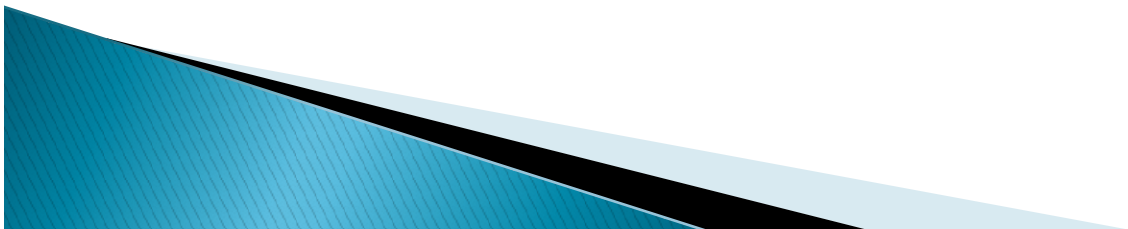
Java Virtual Machine (JVM) & its Architecture

- ▶ What is JVM?

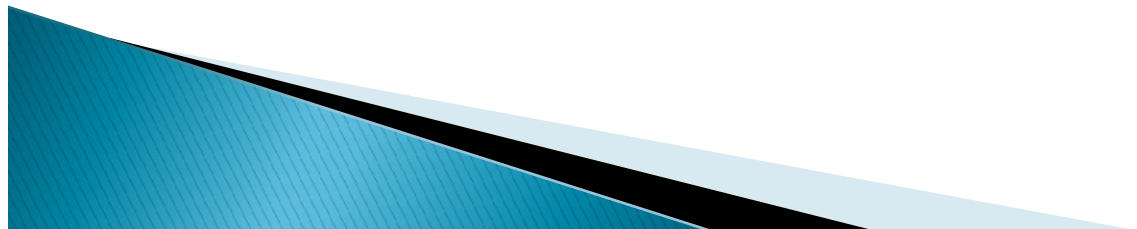
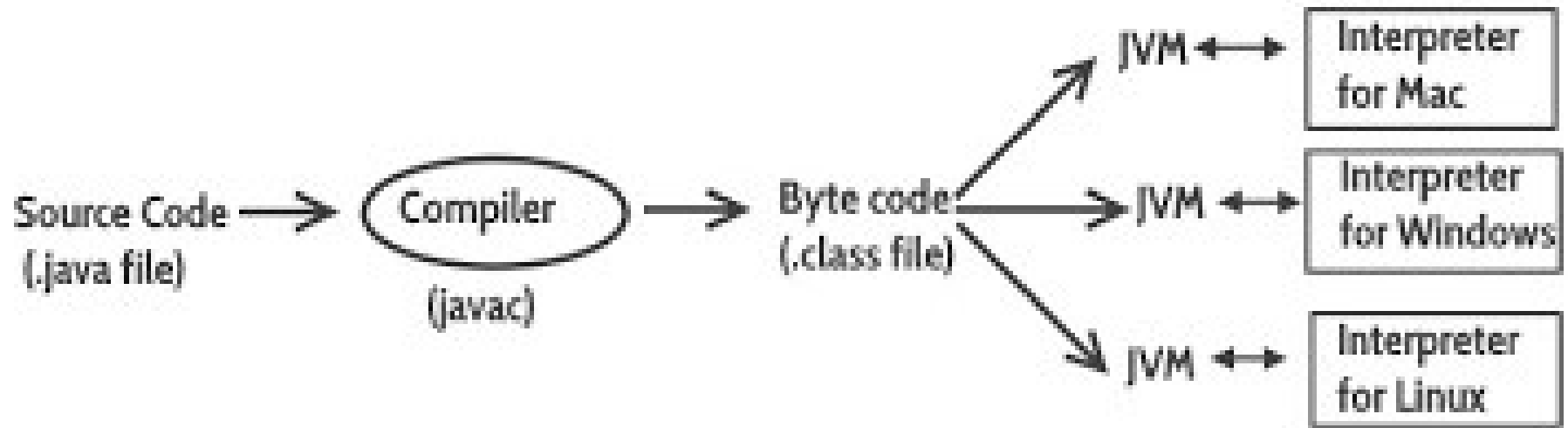
JVM is a run time environment which acts as an interpreter and translates the byte code into object code (machine language). It is a platform-independent execution environment that converts Java byte code into object code and executes it. Java Virtual Machine (JVM) is an engine that provides runtime environment to drive the Java Code or applications. JVM is a part of Java Runtime Environment (JRE). In other programming languages, the compiler produces machine code for a particular system. However, Java compiler produces code for a Virtual Machine known as Java Virtual Machine.

- ▶ How JVM works?

- ▶ Java code is compiled into bytecode. This bytecode gets interpreted on different machines between host system and Java source, Bytecode is an intermediary language.
- ▶ JVM is responsible for allocating memory space.



Java Virtual Machine (JVM) & its Architecture(Cont.)



Java Virtual Machine (JVM) & its Architecture(Cont.)

- ▶ **Source Code**
- ▶ The core program or text written in any computer language (like C, C++, Java, etc.) is called source code. Usually it is a collection of computer instructions written by using any human readable computer language. Source code files have the extension class.
- ▶ **Object Code**
- ▶ The program in the form of machine instructions or binary instructions (i.e., in computer readable form). It is generally produced by compiler/interpreter, but in case of Java, it is produced by JVM.
- ▶ **Byte Code**
- ▶ In Java, when a source code is compiled, it doesn't directly convert into object code, rather it converts into what, is known as byte code. So, a byte code is machine instruction that the Java compiler (Javac) generates.

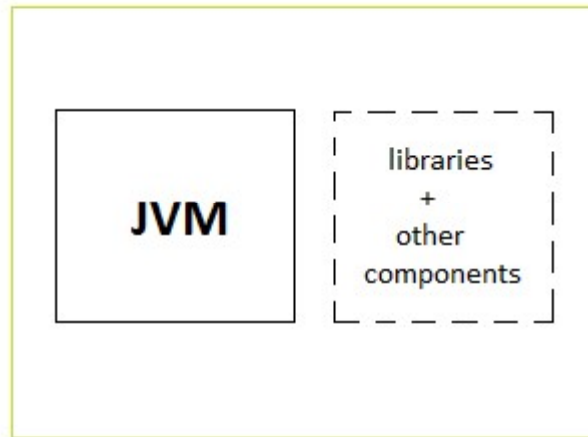


Why is Java both Interpreted and Compiled Language?

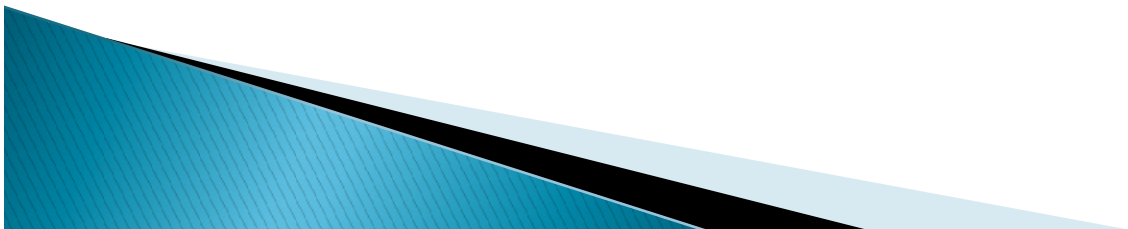
- ▶ Java code is written in .java files (also known as source file), which is compiled by javac, a Java compiler into .class files. Unlike C or C++ compiler, Java compiler doesn't generate native code. These class files contain byte code, which is different than machine or native code.
- ▶ An interpreter is a computer program, which converts each high-level program statement into the machine code. A compiler will convert the code into machine code before program run while Interpreters convert code into machine code when the program is run. In Java, To improve performance, JIT compilers interact with the Java Virtual Machine (JVM) at run time and compile suitable bytecode sequences into native machine code. The JIT compiler is able to perform certain simple optimizations while compiling a series of bytecode to native machine language.

The Just-In-Time (JIT) compiler is an essential part of the IDE

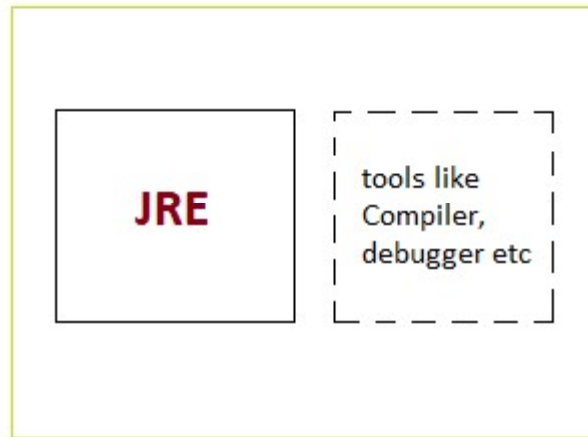
JRE VS JDK



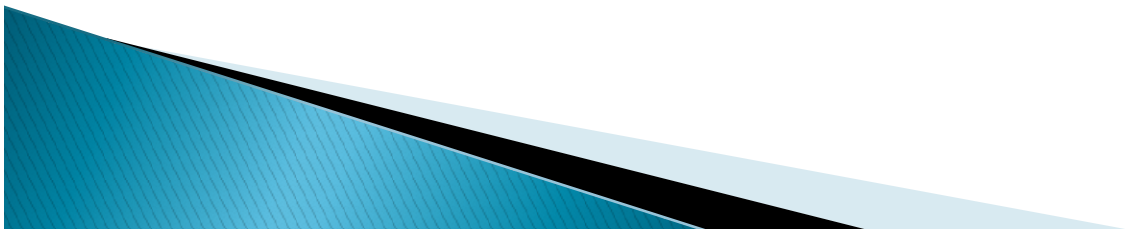
JRE - Java Runtime Environment



JRE VS JDK

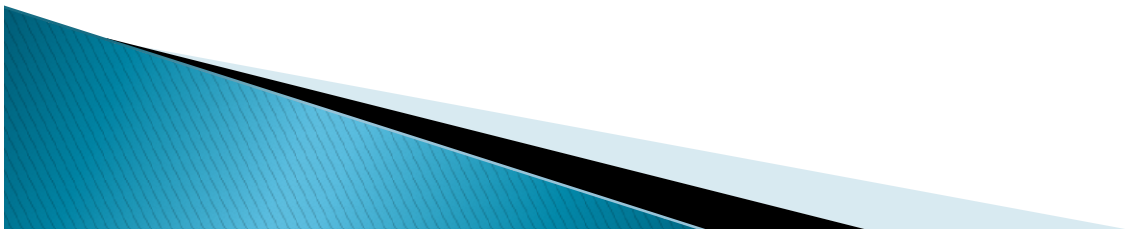


JDK - Java Development Kit



JRE VS JDK

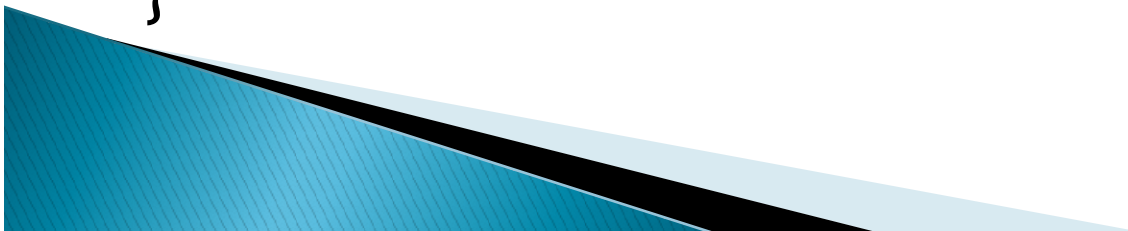
- ▶ **JRE** : The Java Runtime Environment (JRE) provides the libraries, the Java Virtual Machine, and other components to run applets and applications written in the Java programming language. JRE does not contain tools and utilities such as compilers or debuggers for developing applets and applications.
- ▶ **JDK** : The JDK also called Java Development Kit is a superset of the JRE, and contains everything that is in the JRE, plus tools such as the compilers and debuggers necessary for developing applets and applications.



First Java Program

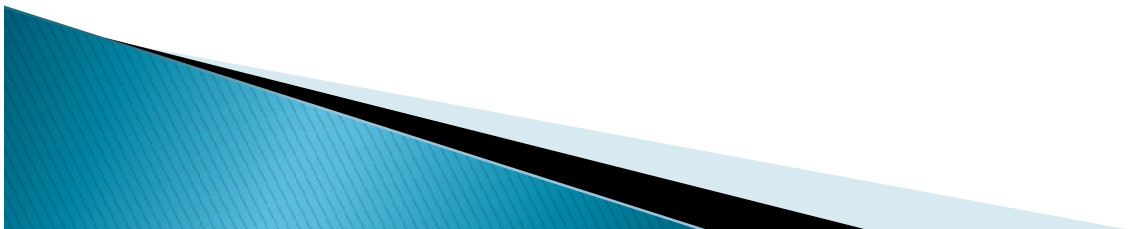
In order to start with basic basic Java programming, let us look at the standard Hello World program.

```
public class MyFirstJavaProgram {  
    public static void main(String []args) {  
        System.out.println("Say Hello World To The  
        World!");  
    }  
}
```



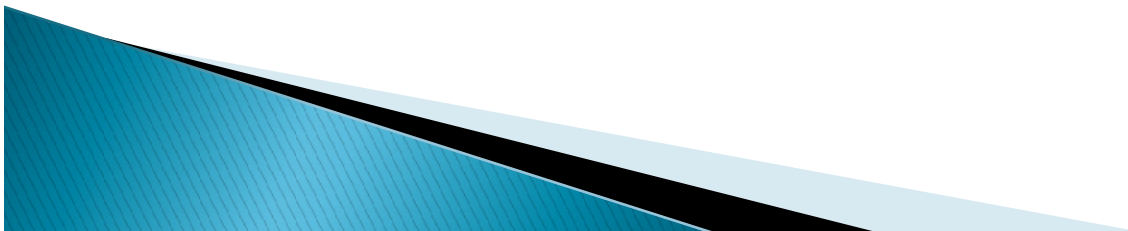
First Java Program

- ▶ Open any text editor and write down the above code in that file.
- ▶ Save the file with a .java extension. For example, you can save the file as MyFirstJavaProgram.java. Name of the file and name of the class containing main() method(Driver method) must be the same.
- ▶ The next step is to to open the command prompt of the system and relocate its reference to the directory in which the file is saved. For instance, if you have saved the file in C:\, then you must take the prompt to the same directory.
- ▶ In order to compile the code, you must type the following:
 - ▶ javac MyFirstJavaProgram.java
 - ▶ If there are no errors, you will automatically be taken to the next line. Successful Compilation leads to the creation of the class file.You can now execute the code using the following command:
 - ▶ java MyFirstJavaProgram
 - ▶ This is the class file which is mentioned here without any extension name. Now You should be able to see the following output on the screen.
 - ▶ Say Hello World To The World!



Structure of Java Program

- ▶ Structure of a java program is the standard format released by Language developer to the Industry programmer. Sun Micro System has prescribed the following structure for the java programmers for developing java application.



Structure of Java Program(Cont.)

package details	→	<code>import java.io.*</code>
<code>class className</code>	→	<code>class Sum</code>
{		
Data members;	→	<code>int a, b, c;</code>
user_defined method;	→	<code>void display();</code>
<code>public static void main(String args[])</code>		
{		
Block of Statements;	→	<code>System.out.println("Hello Java !");</code>
}		
}		

Structure of Java Program(Cont.)

- ▶ A package is a collection of classes, interfaces and sub-packages. A sub package contains collection of classes, interfaces and sub-sub packages etc. `java.lang.*`; package is imported by default and this package is known as default package.
- ▶ class is the keyword used for developing user defined data type and every java program must start with a concept of class.
- ▶ ClassName represents a java valid variable name treated as a name of the class each and every class name in java is treated as user-defined data type.
- ▶ Data member represents either instance or static they will be selected based on the name of the class.
- ▶ User-defined methods represents either instance or static they are meant for performing the operations either once or each and every time.



Structure of Java Program(Cont.)

- ▶ Each and every java program starts execution from the main() method.
and hence main() method is known as program driver. Since main() method of java is not returning any value and hence its return type must be void. Since main() method of java executes only once throughout the java program execution and hence its nature must be static. Since main() method must be accessed by every java programmer and hence whose access specifier must be public.
- ▶ Each and every main() method of java must take array of objects of String.
- ▶ Block of statements represents set of executable statements which are in term calling user-defined methods are containing business-logic.
- ▶ The file naming convention in the java programming is that which-ever class is containing main() method, that class name must be given as a file name with an extension .java.



Main() Method

- ▶ main () method is starting execution block of a java program or any java program start their execution from main method. If any class contain main() method known as main class.

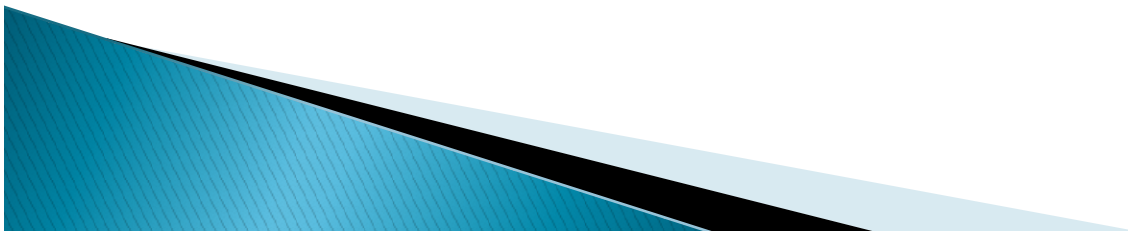
```
public static void main(String args[])
```

```
{
```

```
.....;
```

```
.....;
```

```
}
```



Main() Method(Cont.)

public

public is a keyword in a java language whenever if it is preceded by main() method the scope is available anywhere in the java environment that means main() method can be executed from anywhere. main() method must be accessed by every java programmer and hence its access specifier must be public.

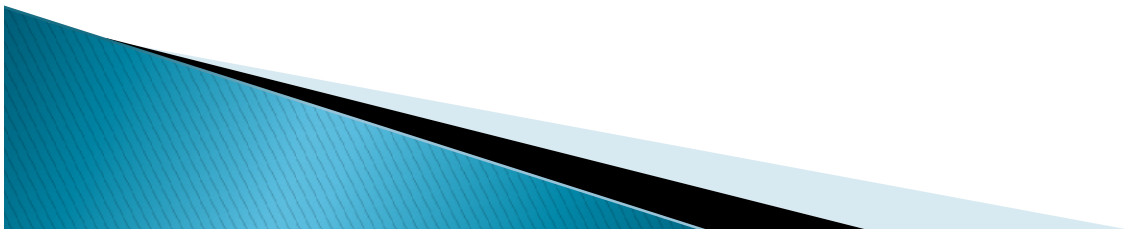
- ▶ **static**

- ▶ **static** is a keyword in java if it is preceded by any class properties for that memory is allocated only once in the program. static method are executed only once in the program. main() method of java executes only once throughout the java program execution and hence it declare must be static.

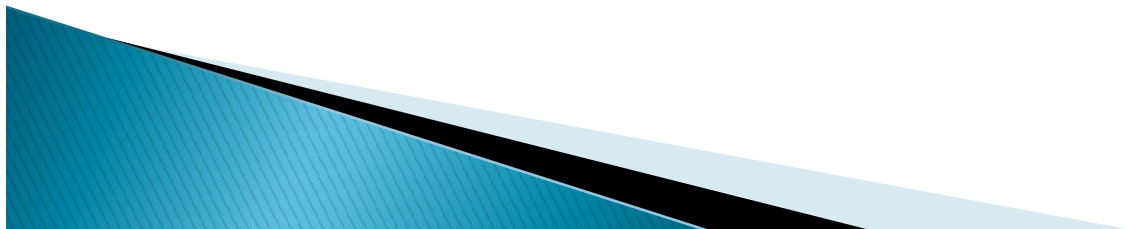
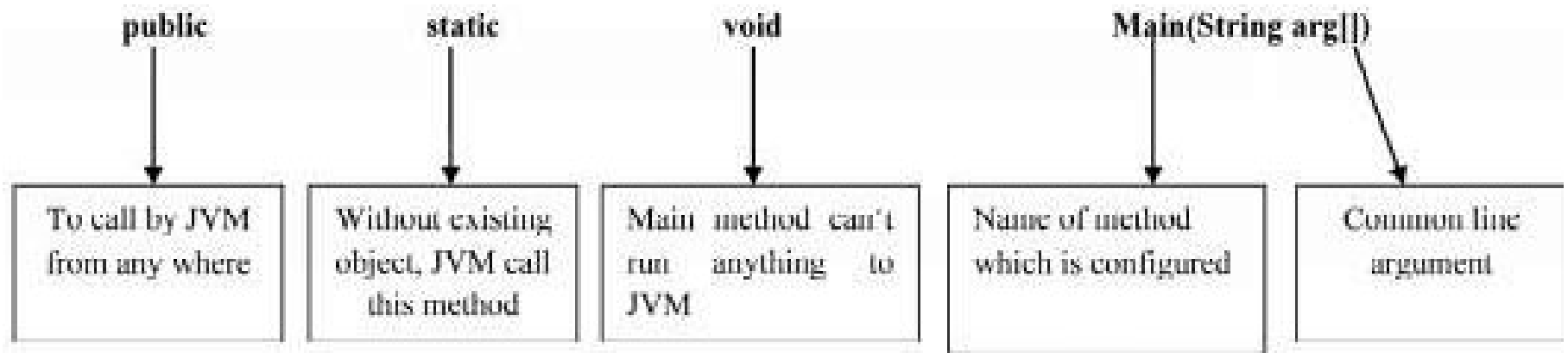


Main() Method(Cont.)

- ▶ **void**
- ▶ **void** is a special datatype also known as no return type, whenever it is preceded by main() method that will be never return any value to the operating system. main() method of java is not returning any value and hence its return type must be void.
- ▶ **String args[]**
- ▶ **String args[]** is a string array used to hold command line arguments in the form of String values.



Main() Method(Cont.)



THANK YOU

