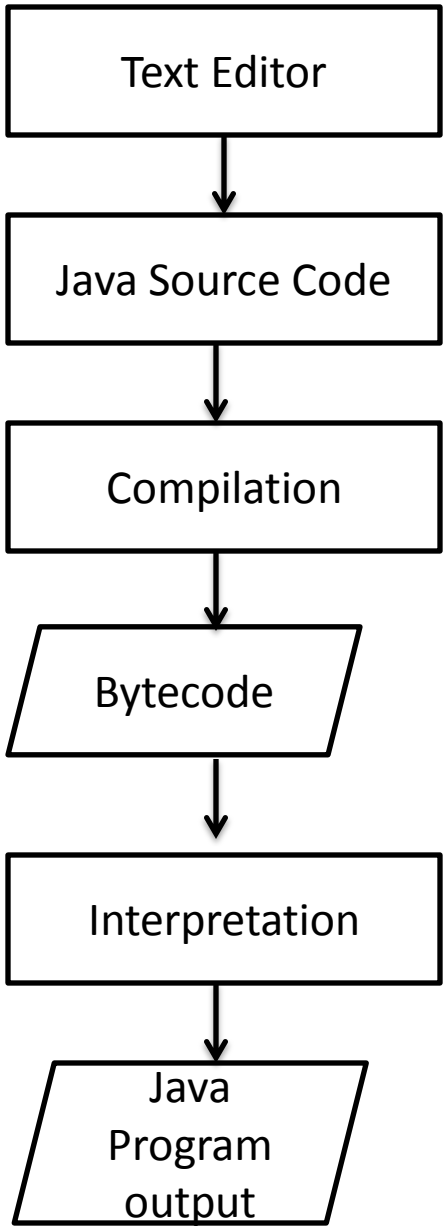
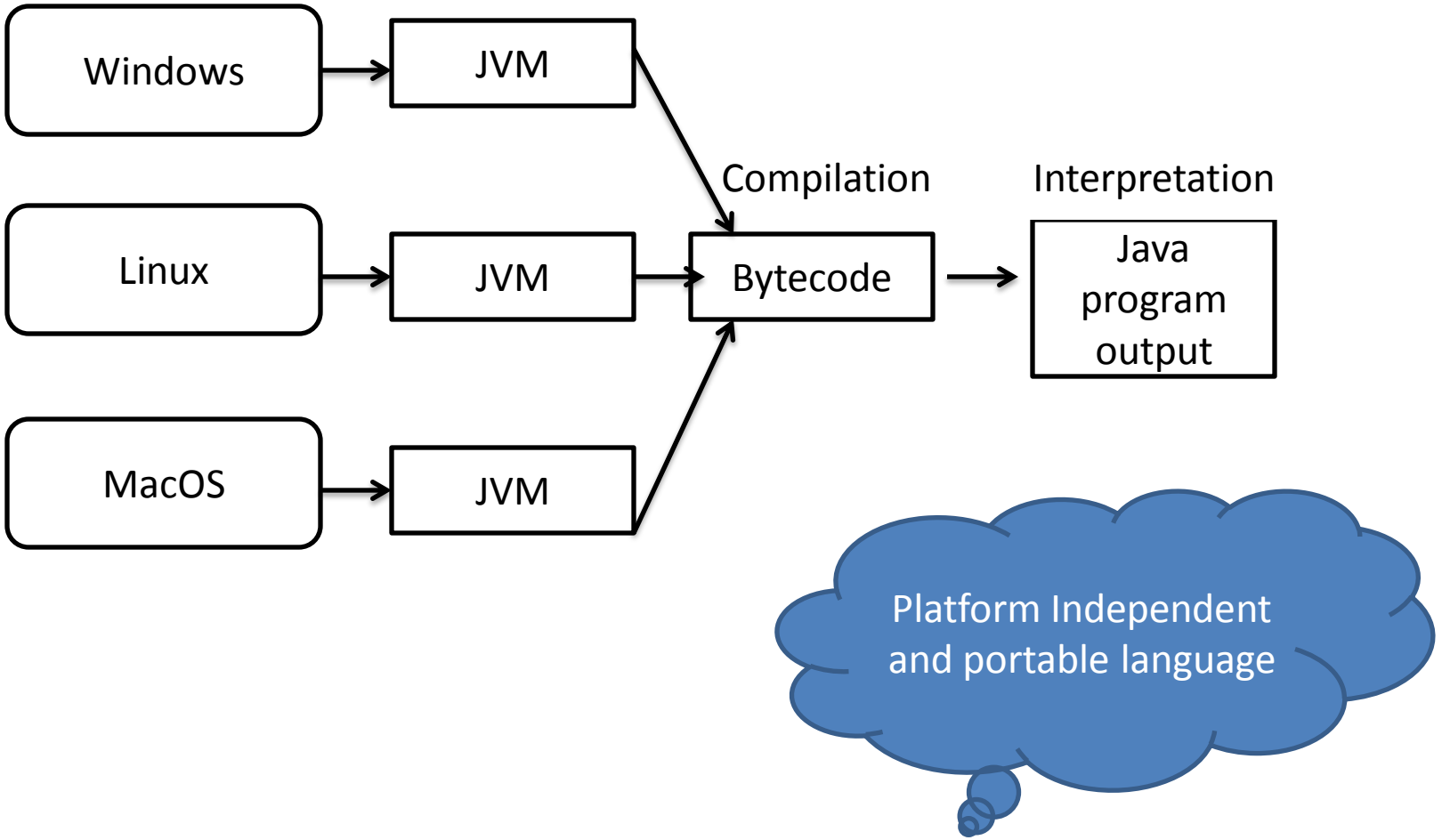
JAVA BASICS

Features of JAVA



Compiled and
interpreted language


Windows
Linux
MacOS
JVM
JVM
JVM
Compilation
Bytecode
Interpretation
Java program output
Platform Independent and portable language

Features

- Compiled and interpreted
- Platform independent and portable
- Object oriented
- Robust and secure
- Distributed
- Familiar, Small and simple
- Multithreaded and Interactive
- High Performance
- Dynamic and extensible

Compiled and interpreted

- A computer language is either compiled or interpreted. Java combines both of these approaches. Hence Java is a two-stage system.

- In the first stage Java compiler translates source into bytecode instructions which is not a machine instruction.

- In second stage Java interpreter generates machine code which that can be executed by the machine that is running the Java program for final output

- Thus Java is both compiled and interpreted language.

Platform independent and portable

- Java programs can be moved from one computer system to another, anywhere and anytime.

- Changes and upgrades in operating systems, processors and system resources will not force any changes in Java program.

- Java ensures portability in two ways.

1. Java compiler generates bytecode instructions that can be implemented on any machine

2. The size of primitive data types are machine independent.

Object - oriented

- Java is truly object-oriented language.
- Object-oriented programming is based upon modeling the world in terms of software components called objects. An object consists of data and operations that can be performed on that data called methods.
- Java comes in extensive set of classes arranged in packages that we can use in programs by inheritance.

Robust and Secure

- Java is a robust language.
- It has strict compile time and run time checking for data types.
- It is garbage collected language solving memory management problem.
- Java supports exception handling that captures error and eliminates risk of crashing the system.
- Java do not support pointers so that program cannot gain access to memory locations without proper authorization.
- Java ensures that no viruses are communicated with an applet.

Distributed

- Java is designed as a distributed language for creating applications on networks.

- Java has a large library of classes for communicating using the Internet's TCP/IP protocol suite, including protocols such as HTTP and FTP.

- This enables multiple programmers at multiple remote locations to collaborate and work together on a single project.

Simple,Small and Familiar

- Java is small and simple language.
- Java does not support
  - Use of pointers
  - Preprocessor header files
  - Goto statement
  - Operator overloading
  - Multiple inheritance

Multithreaded and interactive

- Multiple thread means multiple task.
- Java supports multithreaded programs. This means that we do not have to wait for an application to finish to begin another one.
- This means that we can play an audio file while browsing on the net and at the same time download an applet file
- Java supports multiple process synchronization and supports interactive system also.

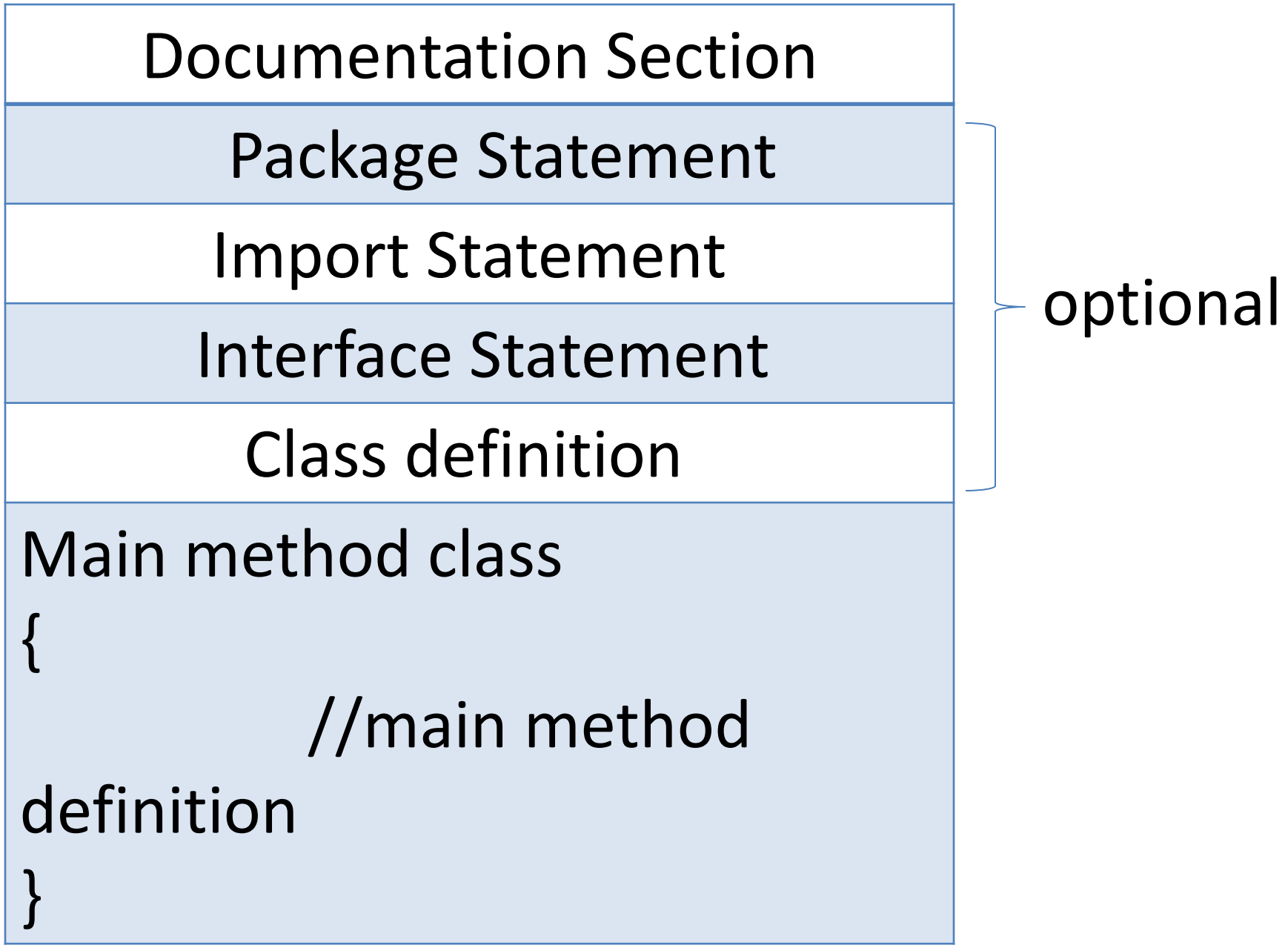
High Performance

- Java performance is impressive due to the use of intermediate bytecode.
- Java architecture is designed to reduce overheads during runtime.
- Support for multithreading enhances overall speed of execution of Java program

Dynamic and extensible

- Java is dynamic language.
- can dynamically link new class libraries, method and objects
- Java program supports functions written in other languages such as C / C++. These functions are called as native methods.
- Native method are linked dynamically at runtime.

Structure of Java Program


Documentation Section
Package Statement
Import Statement
Interface Statement
Class definition
optional
Main method class
{
//main method
definition
}

```
class Sample
{
public static void main(String args[])
{
System.out.println("Hello Java");
}
}
```

- The first line,

class Sample

Declares the class. As Java is true-object oriented programming language, everything in Java must be placed inside class.

“Sample” is a Java identifier that specifies name of the class.

- The main line is

  public static void main(String args[ ])

  1. Public – is an access specifier that declares the main method is unprotected and is accessible to all other classes.

  2. Static – it declares that this method belongs to the entire class and not as a part of any particular object

  3. Void- it states that main() method does not return any value.

- main():-It is the starting point for the interpreter to begin the execution of the program.

A Java application can have any number of classes but only one of them must include main method to initiate execution.

- Lets see an example…

```
class A{

}
class B{

}
class C{

}
class D
{
        public   static void main(String args[])
        {

        }
}
```

- String args[]- it declares parameter named args, which contains an array of objects of type String.

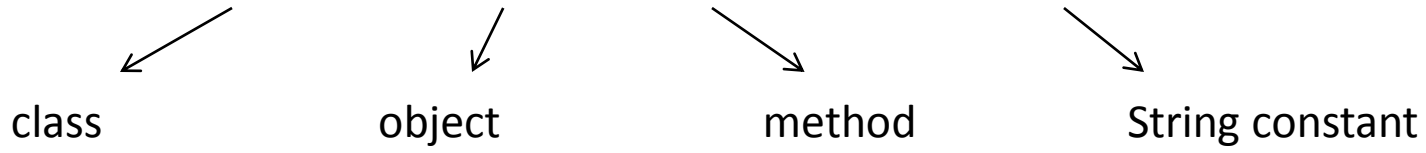
- The output line ,

System.out.println(“Hello World”)

The **println()** method is member of **out** object which is static data member of **System** class.

It is similar to **printf** in C and **cout** in C++. So

System.out.println(“Hello World”)



- Explanation of public static void main
- Exexcution steps from command prompt screen shot.
- Java support system
- Java Environment
- Application Programming Interface.

Lets see the execution !!!

- Save the file with extension Sample.java
- Open command prompt and go to the location where you have saved the file.
- First compile the file using javac (java compiler)
- Then run the program using java (java interpreter)