

MODULE 1

CHAPTER 2 INTRODUCTION TO JAVA

Prepared By EBIN PM, AP, IESCE

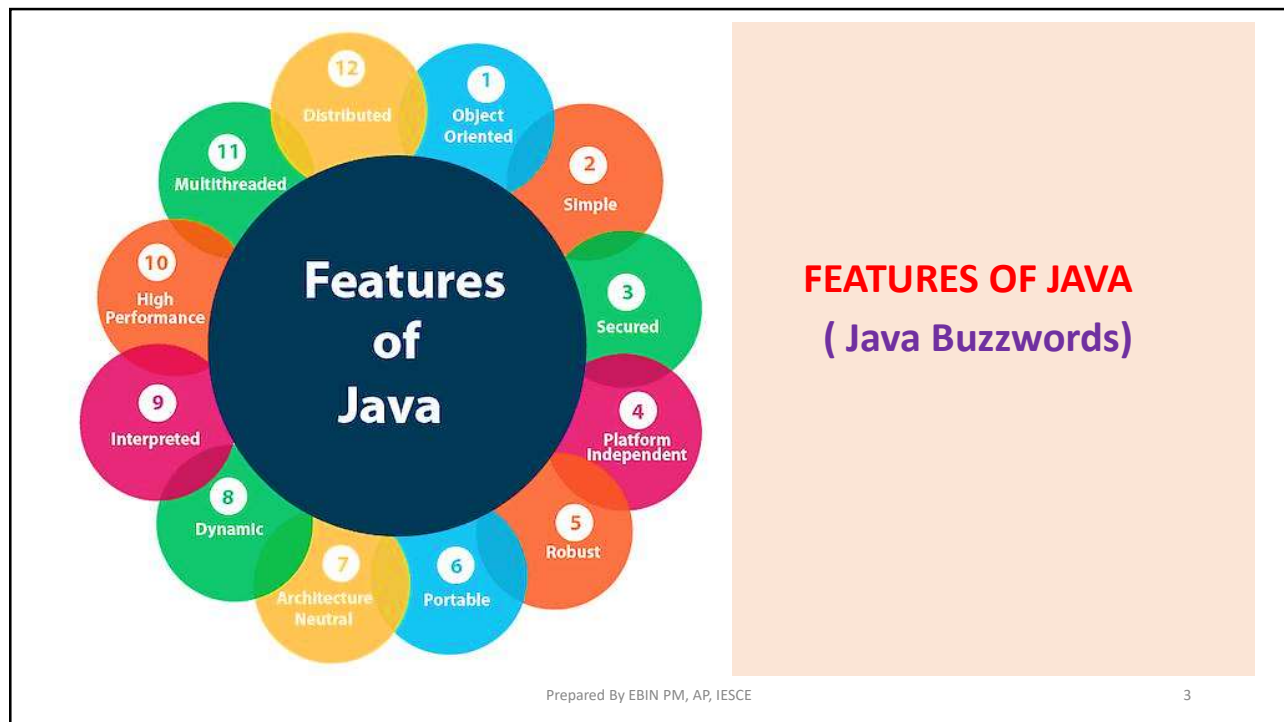
1

JAVA

- Java is a powerful general-purpose , Object Oriented programming language developed by Sun Micro System of USA in 1991.
- Development team members are James Gosling, Patrick Naughton, Chris Warth, Ed Frank, and Mike Sheridan
- First name of Java is “Oak,” but was renamed “Java” in 1995.
- Java derives much of its character from C and C++.
- Java Changed the Internet by simplifying web programming
- Java innovated a new type of networked program called the **applet**

Prepared By EBIN PM, AP, IESCE

2



❖ **JAVA RUNTIME ENVIRONMENT (JRE)**

- A software program needs an environment to run .
- The runtime environment loads class files and ensures there is access to memory and other **system resources** to run them.
- Java Runtime Environment provides the minimum requirements for executing a Java application programs.
- JRE is an **installation package** which provides environment to **only run(not develop)** the java program(or application)onto your machine.
- JRE is only used by them who only wants to run the Java Programs i.e. end users of your system. JRE can be view as a **subset of JDK**.

Prepared By EBIN PM, AP, IESCE

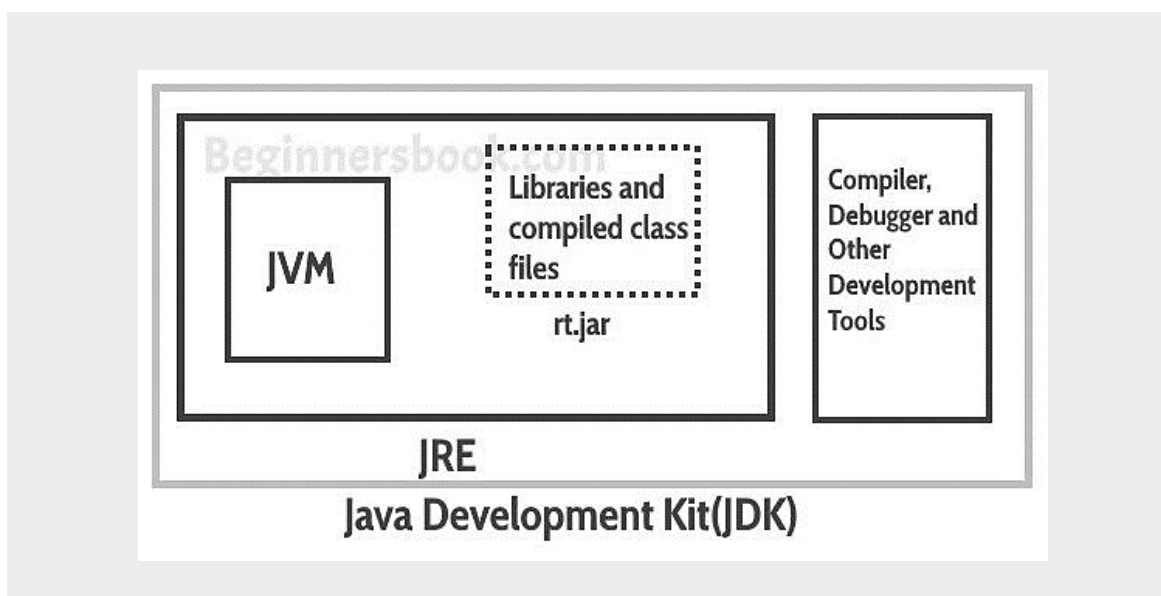
4

❖ JAVA DEVELOPMENT KIT (JDK)

- The Java Development Kit (JDK) is a software development environment used for developing and executing Java applications and applets
- It includes the Java Runtime Environment (JRE), an interpreter/loader (Java), a compiler (javac), an archiver (jar), a documentation generator (Javadoc) and other tools needed in Java development.
- JDK is only used by Java Developers.

Prepared By EBIN PM, AP, IESCE

5



Prepared By EBIN PM, AP, IESCE

6

❖ JAVA VIRTUAL MACHINE (JVM)

- JVM is a program which provides the runtime environment to execute Java programs. Java programs cannot run if a supporting JVM is not available.
- JVM is a virtual machine that resides in the real machine (your computer) and the **machine language for JVM is byte code**.
- The Java compiler generate byte code for JVM rather than different machine code for each type of machine.
- **JVM executes the byte code generated by compiler** and produce output.
- JVM is the one that makes java platform independent.

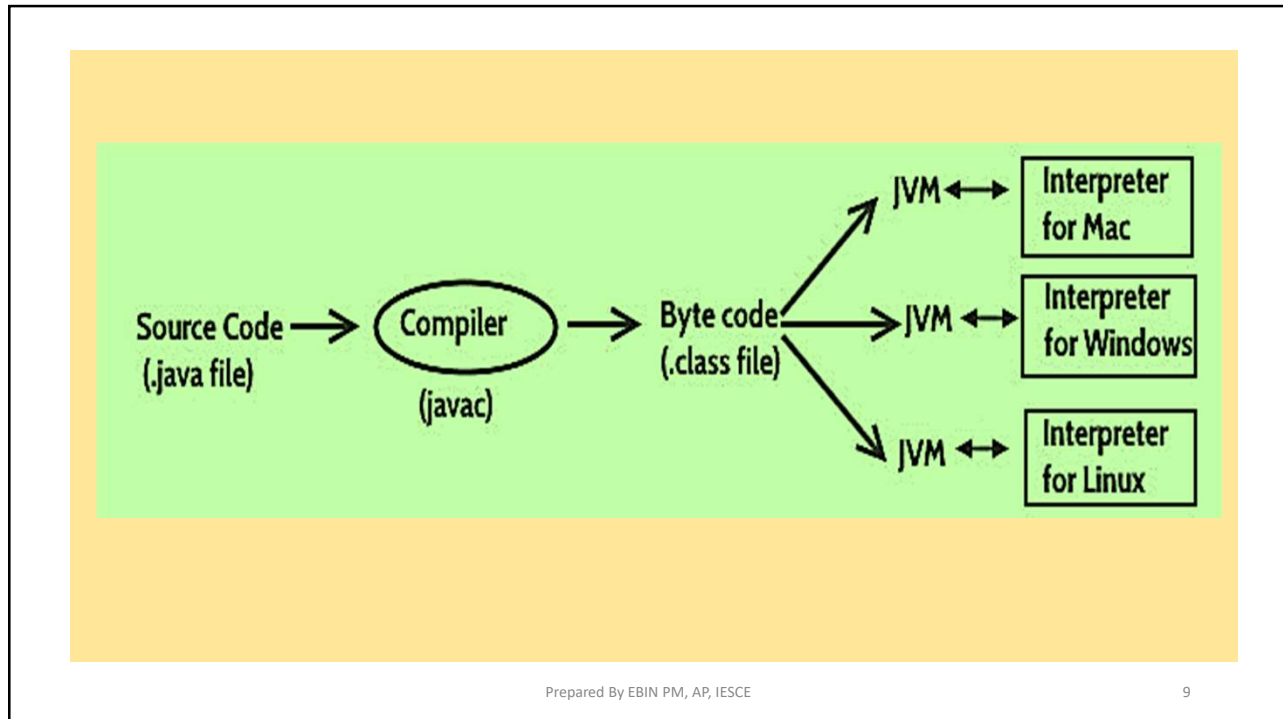
Prepared By EBIN PM, AP, IESCE

7

- The **primary function** of JVM is to **execute the byte code** produced by compiler
- The JVM doesn't understand Java source code, that's why we need to have **javac** compiler
- Java compiler (javac) compiles *.java files to obtain *.class files that contain the byte codes understood by the JVM.
- JVM makes java portable (write once, run anywhere).
- Each operating system has different JVM, however the output they produce after execution of byte code is same across all operating systems.

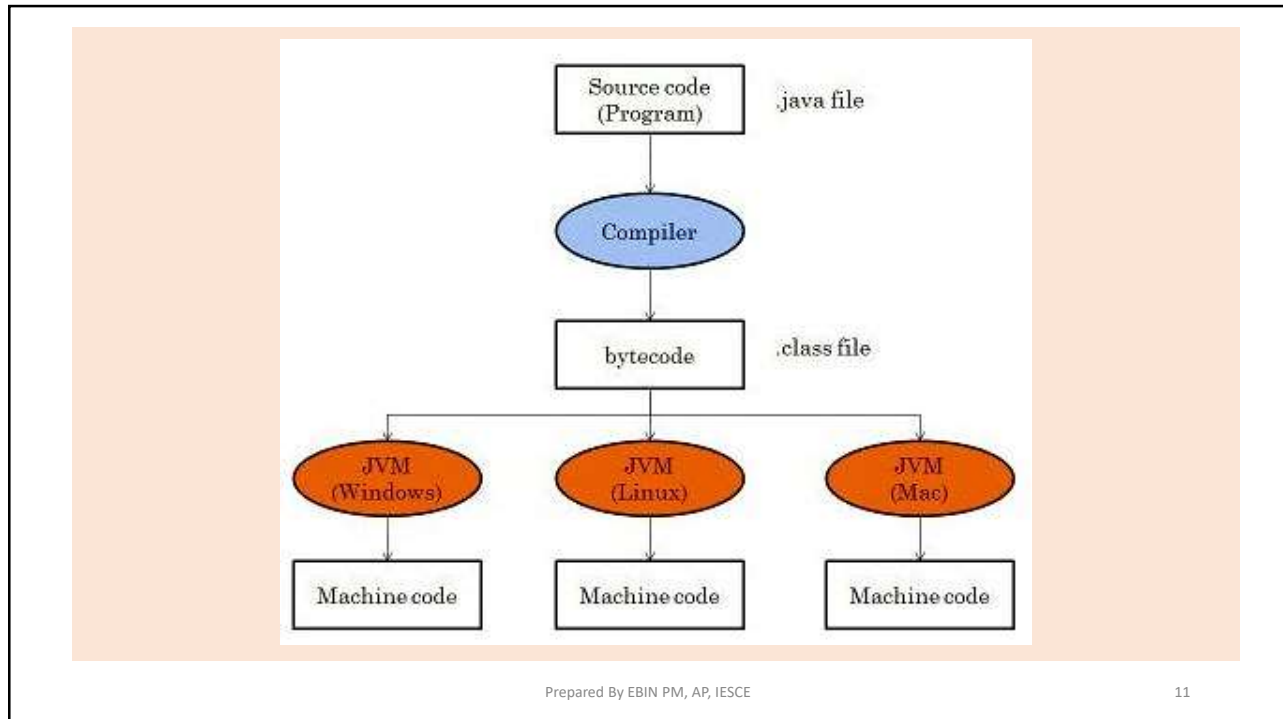
Prepared By EBIN PM, AP, IESCE

8



❖ BYTE CODE

- Java byte code is the instruction set for the Java Virtual Machine
- It is the machine code in the form of a .class file.
- Byte code is a machine independent code
- It is not completely a compiled code but it is an intermediate code somewhere in the middle which is later interpreted and executed by JVM.
- Byte code is a machine code for JVM.
- Byte code implementation makes Java a platform- Independent language.



❖ JAVA COMPILER

- Java is compiled language. But it is very different from traditional compiling in the way that after compilation source code is converted to byte code.
- **Javac** is the most popular Java compiler
- Java has a virtual machine called JVM which then converts byte code to target code of machine on which it is run.
- JVM performs like an interpreter. It doesn't do it alone, though. It has its own compiler to convert the byte code to machine code. This compiler is called **Just In Time** or **JIT compiler**.

❖ JAVA APPLET

- An *applet* is a special kind of Java program that is designed to be transmitted over the Internet and automatically executed by a Java-compatible web browser
- It runs inside the web browser and works at client side
- Applets are used to make the web site more dynamic and entertaining
- Applets are not stand-alone programs. Instead, they run within either a web browser or an applet viewer. JDK provides a standard applet viewer tool called applet viewer.
- In general, execution of an applet does not begin at `main()` method.

Prepared By EBIN PM, AP, IESCE

13



Lifecycle of Java Applet

Prepared By EBIN PM, AP, IESCE

14

Java Applet vs Java Application

Java Application	Java Applet
Java Applications are the stand-alone programs which can be executed independently	Java Applets are small Java programs which are designed to exist within HTML web document
Java Applications must have main() method for them to execute	Java Applets do not need main() for execution
Java Applications just needs the JRE	Java Applets cannot run independently and require API's
Java Applications do not need to extend any class unless required	Java Applets must extend java.applet.Applet class
Java Applications can execute codes from the local system	Java Applets Applications cannot do so
Java Applications has access to all the resources available in your system	Java Applets has access only to the browser-specific services

Prepared By EBIN PM, AP, IESCE

15

JAVA BUZZWORDS

➤ Simple

- It's simple and easy to learn if you already know the basic concepts of Object Oriented Programming.
- C++ programmer can move to JAVA with very little effort to learn.
- Java syntax is based on C++
- Java has removed many complicated and rarely-used features, for example, explicit pointers, operator overloading, etc.

Prepared By EBIN PM, AP, IESCE

16

➤ Object oriented

- Java is true object oriented language. Everything in Java is an object.
- All program code and data reside within objects and classes.
- Java comes with an extensive set of classes, arranged in packages that can be used in our programs through inheritance.

➤ Distributed

- Java is designed for distributed environment of the Internet. Its used for creating applications on networks
- Java enables multiple programmers at multiple remote locations to collaborate and work together on a single project.

Prepared By EBIN PM, AP, IESCE

17

➤ Compiled and Interpreted

- Usually a computer language is either compiled or Interpreted. Java combines both this approach and makes it a two-stage system.
- Compiled : Java enables creation of a cross platform programs by compiling into an intermediate representation called Java Byte code.
- Interpreted : Byte code is then interpreted, which generates machine code that can be directly executed by the machine that provides a Java Virtual machine.

Prepared By EBIN PM, AP, IESCE

18

➤ Robust

- It provides many features that make the **program execute reliably** in variety of environments.
- Java is a **strictly typed language**. It checks code both at compile time and runtime.
- Java takes care of all memory management problems with **garbage-collection**.
- Java, with the help of **exception handling** captures all types of serious errors and eliminates any risk of crashing the system.

Prepared By EBIN PM, AP, IESCE

19

➤ Secure

- Java provides a **“firewall”** between a networked application and your computer.
- When a Java Compatible Web browser is used, **downloading can be done safely** without fear of viral infection or malicious intent.
- Java achieves this protection by confining a Java program to the java execution environment and not allowing it to access other parts of the computer.

➤ Architecture Neutral

- Java language and Java Virtual Machine helped in achieving the goal of **“write once; run anywhere, any time, forever.”**
- Changes and upgrades in operating systems, processors and system resources will not force any changes in Java Programs.

Prepared By EBIN PM, AP, IESCE

20

➤ Portable

- Java is portable because it facilitates you to **carry the Java byte code to any platform**. It doesn't require any implementation.
- Java Provides a way to download programs dynamically to all the various types of platforms connected to the Internet.

➤ High Performance

- Java performance is high because of the use of byte code.
- The byte code can be easily translated into native machine code.

Prepared By EBIN PM, AP, IESCE

21

➤ Multithreaded

- Multithreaded Programs **handled multiple tasks simultaneously**, which was helpful in creating interactive, networked programs.
- Java run-time system comes with tools that support multiprocess synchronization used to construct smoothly interactive systems

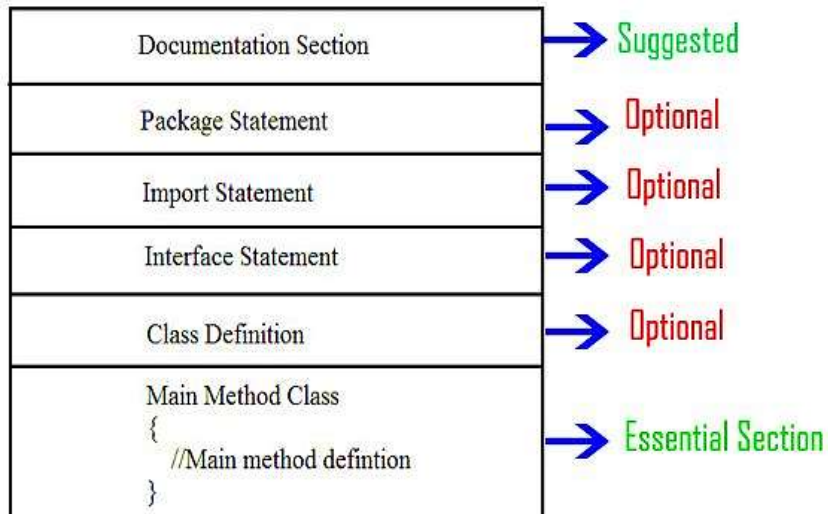
➤ Dynamic

- Java is capable of linking in new class libraries, methods, and objects.
- It supports functions from native languages (the functions written in other languages such as C and C++).
- It supports dynamic loading of classes. It means classes are loaded on demand

Prepared By EBIN PM, AP, IESCE

22

JAVA PROGRAM STRUCTURE



Prepared By EBIN PM, AP, IESCE

23

➤ Documentation Section

- You can write a comment in this section. It helps to understand the code. These are optional
- It is used to improve the readability of the program.
- The compiler ignores these comments during the time of execution
- There are three types of comments that Java supports
 - Single line Comment //This is single line comment
 - Multi-line Comment /* this is multiline comment.
and support multiple lines*/
 - Documentation Comment /** this is documentation cmnt*/

Prepared By EBIN PM, AP, IESCE

24

➤ Package Statement

- We can create a package with any name. A package is a **group of classes** that are defined by a name.
- That is, if you want to declare many classes within one element, then you can declare it within a package
- It is an optional part of the program, i.e., if you do not want to declare any package, then there will be no problem with it, and you will not get any errors.
- Package is declared as: `package package_name;`
Eg: `package mypackage;`

Prepared By EBIN PM, AP, IESCE

25

➤ Import Statement

- If you want to use a class of another package, then you can do this by importing it directly into your program.
- Many predefined classes are stored in packages in Java
- We can import a specific class or classes in an import statement.

Examples:

```
import java.util.Date; //imports the date class
```

```
import java.applet.*; /*imports all the classes from the java  
applet package*/
```

Prepared By EBIN PM, AP, IESCE

26

➤ Interface Statement

- This section is used to specify an interface in Java
- Interfaces are like a class that includes a group of method declarations
- It's an optional section and can be used when programmers want to implement multiple inheritances within a program.

➤ Class Definition

- A Java program may contain several class definitions.
- Classes are the main and essential elements of any Java program.
- A class is a collection of variables and methods

Prepared By EBIN PM, AP, IESCE

27

➤ Main Method Class

- The main method is from where the **execution actually starts** and follows the order specified for the following statements
- Every Java stand-alone program requires the main method as the starting point of the program.
- This is an essential part of a Java program.
- There may be many classes in a Java program, and only one class defines the main method
- Methods contain data type declaration and executable statements.

Prepared By EBIN PM, AP, IESCE

28

A simple java program to print hello world

```
public class Hello
{
    //main method declaration
    public static void main(String[] args)
    {
        System.out.println("hello world");
    }
}
```

Prepared By EBIN PM, AP, IESCE

29

- **public class Hello** - This creates a class called Hello. We should make sure that the **class name starts with a capital letter**, and the public word means it is accessible from any other classes.
- **Braces** - The curly brackets are used to group all the commands together
- **public static void main**
 - When the main method is declared **public**, it means that it can be used outside of this class as well.
 - The word **static** means that we want to access a method without making its objects
 - The word **void** indicates that it does not return any value. The main is declared as void because it does not return any value.
 - *main* is a method; this is a starting point of a Java program.

Prepared By EBIN PM, AP, IESCE

30

➤ **String[] args**

It is an array where each element is a string, which is named as args. If you run the Java code through a console, you can pass the input parameter. The main() takes it as an input.

➤ **System.out.println();**

- This statement is used to print text on the screen as output
- system is a predefined class, and out is an object of the PrintWriter class defined in the system
- The method println prints the text on the screen with a new line.
- We can also use print() method instead of println() method. All Java statement ends with a semicolon.

Prepared By EBIN PM, AP, IESCE

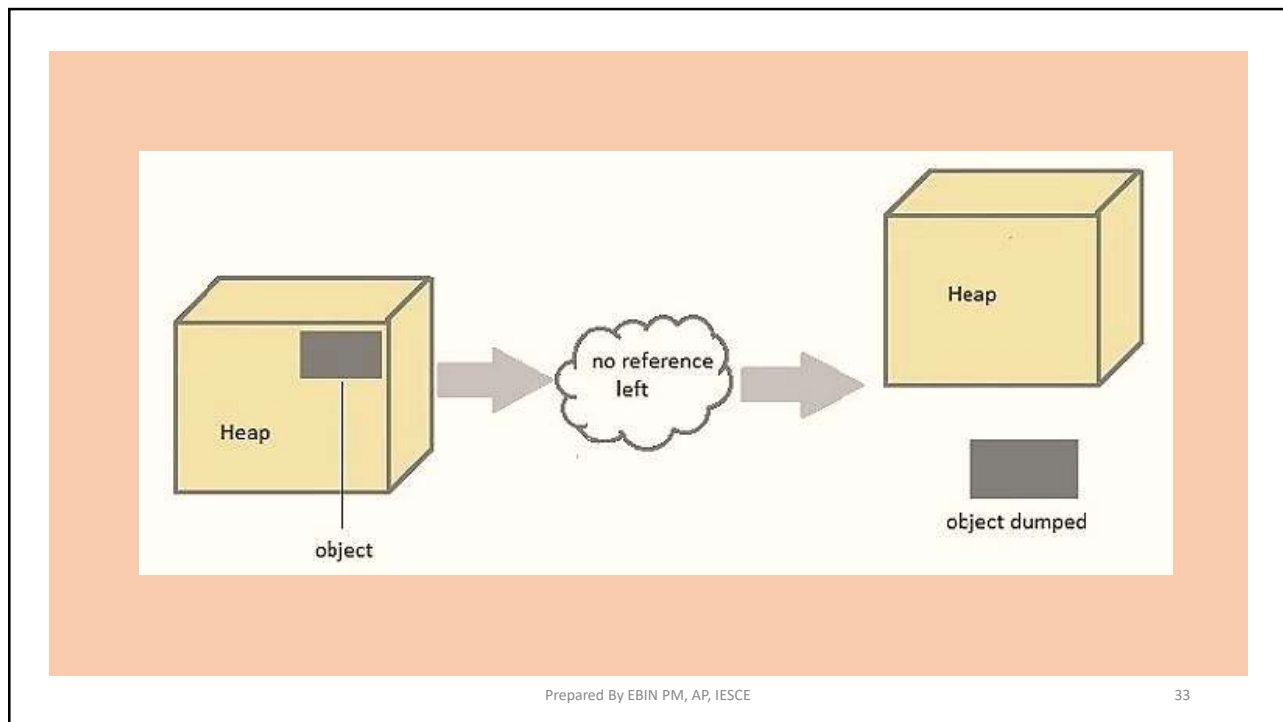
31

Garbage Collection in Java (A process of releasing unused memory)

- When JVM starts up, it creates a heap area which is known as runtime data area. This is where all the objects (instances of class) are stored
- Since this area is limited, it is required to manage this area efficiently by removing the objects that are no longer in use.
- The process of removing unused objects from heap memory is known as Garbage collection and this is a part of memory management in Java.
- Languages like C/C++ don't support automatic garbage collection, however in java, the garbage collection is automatic.

Prepared By EBIN PM, AP, IESCE

32



- In java, **garbage** means **unreferenced objects**.
- Main objective of Garbage Collector is to free heap memory by destroying unreachable objects.
- Unreachable objects : An object is said to be unreachable iff it doesn't contain any reference to it.
- Eligibility for garbage collection : An object is said to be eligible for GC(garbage collection) iff it is unreachable.
- **finalize() method** – This method is invoked each time before the object is garbage collected and it perform cleanup processing.
- The Garbage collector of JVM collects only those objects that are created by new keyword. So if we have created any object without new, we can use finalize method to perform cleanup processing

Request for Garbage Collection

- We can request to JVM for garbage collection however, it is upto the JVM when to start the garbage collector.
- Java `gc()` method is used to **call garbage collector explicitly**.
- However `gc()` method does not guarantee that JVM will perform the garbage collection.
- It only request the JVM for garbage collection. This method is present in **System and Runtime class**.

Prepared By EBIN PM, AP, IESCE

35

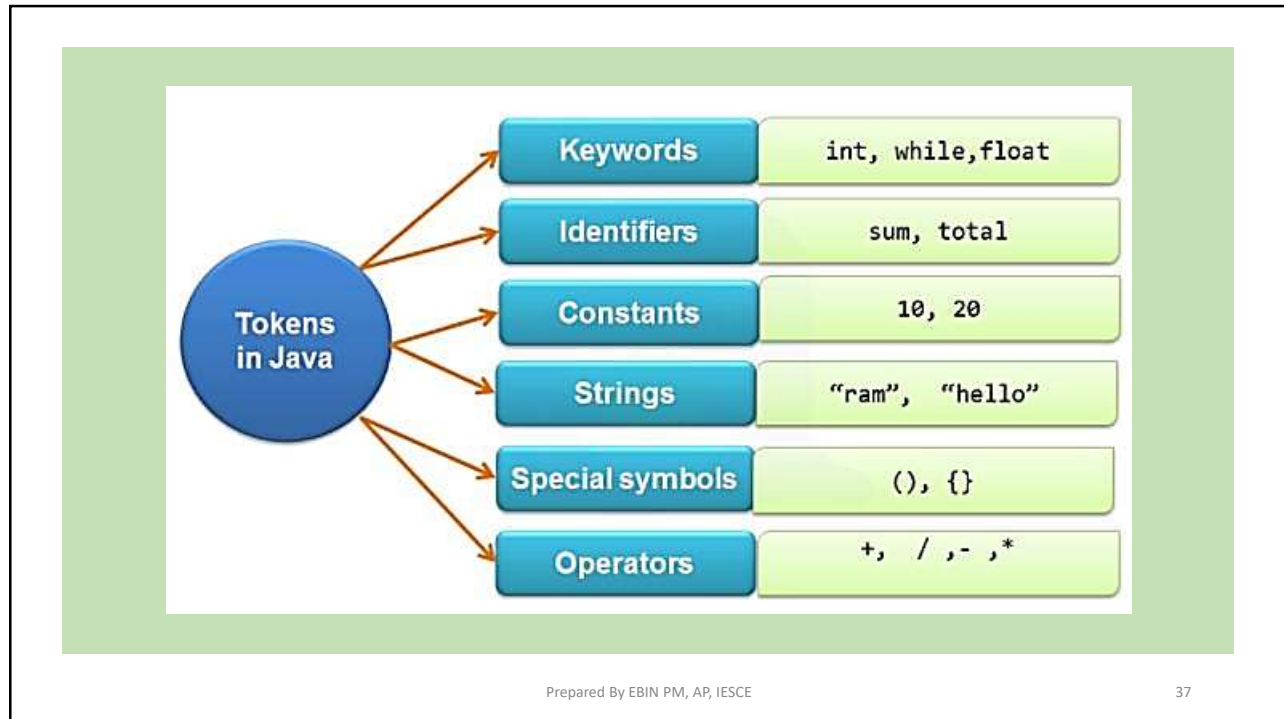
Java Lexical Issues (Java Tokens)

❖ TOKENS

- Java Tokens are the smallest individual building block or smallest unit of a Java program
- Java program is a collection of different types of tokens, comments, and white spaces.

Prepared By EBIN PM, AP, IESCE

36



➤ Keywords

- A keyword is a **reserved word**. You cannot use it as a variable name, constant name etc.
- The meaning of the keywords has already been described to the java compiler. These meaning cannot be changed.
- Thus, the keywords cannot be used as variable names because that would try to change the existing meaning of the keyword, which is not allowed.
- Java language has reserved **50 words** as keywords

Keywords in Java				
abstract	default	if	private	this
assert	do	implements	protected	throw
boolean	double	import	public	throws
break	else	instanceof	return	transient
byte	enum	int	short	try
case	extends	interface	static	void
catch	final	long	strictfp	volatile
char	finally	native	super	while
class	float	new	switch	
continue	for	package	synchronized	

Prepared By EBIN PM, AP, IESCE 39

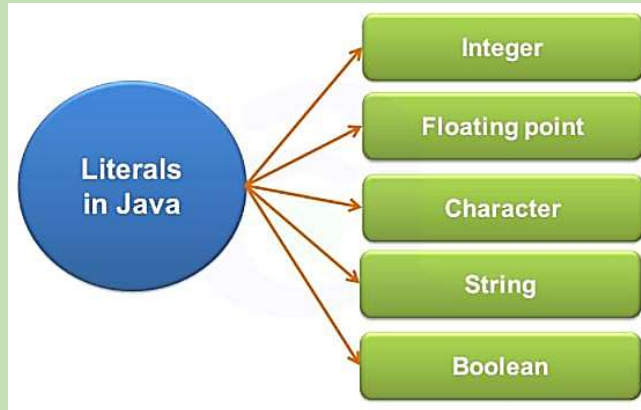
➤ **Identifiers**

- Identifiers are the names of variables, methods, classes, packages and interfaces
- Identifier must follow some rules.
 - ✓ All identifiers must start with either a letter(a to z or A to Z) or currency character(\$) or an underscore.
 - ✓ They must not begin with a digit
 - ✓ After the first character, an identifier can have any combination of characters.
 - ✓ A Java keywords cannot be used as an identifier.
 - ✓ Identifiers in Java are case sensitive, foo and Foo are two different identifiers.
 - ✓ They can be any length Eg: int a; char name;

Prepared By EBIN PM, AP, IESCE 40

➤ Constants or Literals

- Constants are fixed values of a particular type of data, which cannot be modified in a program.
- Java language specifies five major type of literals.



Prepared By EBIN PM, AP, IESCE

41

Eg: Integer literal : 100

Floating-point literal : 98.6

Character literal : 's'

String literal : "sample"

➤ Comments

Comment type	Meaning
// comment	Single-line comments
/* comment */	Multi-line comments
/** documentation */	Documentation comments

Prepared By EBIN PM, AP, IESCE

42

➤ String

- In java, string is basically an object that represents sequence of char values.
- An array of characters works same as java string.

```
Eg: char[] ch = {'a','t','n','y','l','a'};
     String s = "atnyla";
```

- Java String class provides a lot of methods to perform operations on string such as compare(), concat(), equals(), split(), length(), replace(), compareTo(), intern(), substring() etc.

Prepared By EBIN PM, AP, IESCE

43

➤ Special symbol

,	<	>	.	_
()	;	\$:
%	[]	#	?
'	&	{	}	"
^	!	*	/	
-	\	~	+	

Prepared By EBIN PM, AP, IESCE

44

Brackets[] : Opening and closing brackets are used as array element reference. These indicate single and multidimensional subscripts.

Parentheses() : These special symbols are used to indicate function calls and function parameters.

Braces{ } : These opening and ending curly braces mark the start and end of a block of code containing more than one executable statement.

semicolon ; : It is used to separate more than one statements like in for loop is separates initialization, condition, and increment.

comma , : It is an operator that essentially invokes something called an initialization list.

asterisk * : It is used for multiplication.

assignment operator = : It is used to assign values.

Period . : Used to separate package names from subpackages and classes

Prepared By EBIN PM, AP, IESCE

45

➤ Operators

- An operator is a symbol that takes one or more arguments and operates on them to produce a result.
- Unary Operator
- Arithmetic Operator
- shift Operator
- Relational Operator
- Bitwise Operator
- Logical Operator
- Ternary Operator
- Assignment Operator

Prepared By EBIN PM, AP, IESCE

46

➤ **Whitespace**

- Java is a free-form language. This means that you do not need to follow any special indentation rules
- White space in Java is used to separate tokens in the source file. It is also used to improve readability of the source code.

Eg: `int i = 0;`

- White spaces are required in some places. For example between the int keyword and the variable name.
- In java whitespace is a space, tab, or newline