**Practical No.1**

**Name**: Anuj Rajendra Mane

**Div**: SY A

**Roll No**: 65 **Batch**: A3

**PRN** **No**: 2162701242054

**Title: Introduction to java**

The history of java starts from Green Team. Java team members (also known as Green Team), initiated a revolutionary task to develop a language for digital devices such as set-top boxes, television sets.

For the green team members, it was an advance concept at that time. But, it was suited for internet programming. Later, Java technology as incorporated by Netscape.

Currently, Java is used in internet programming, mobile devices, games, e-business solutions etc. There are given the major points that describes the history of java.

1) James Gosling, Mike Sheridan, and Patrick Naughton initiated the Java language project in June 1991. The small team of sun engineers called Green Team.

2) Originally designed for small, embedded systems in electronic appliances like set top boxes.

3) Firstly, it was called "Green talk" by James Gosling and file extension was.gt.

4) After that, it was called Oak and was developed as a part of the Green project.

**Java Version History**

There are many java versions that has been released. Current stable release of Java is Java SE 8.

1. JDK Alpha and Beta (1995)

2. JDK 1.0 (23rd Jan, 1996)

3. JDK 1.1 (19th Feb, 1997)

4. J2SE 1.2 (8th Dec, 1998)

5. J2SE 1.3 (8th May, 2000)

6. J2SE 1.4 (6th Feb, 2002)

7. J2SE 5.0 (30th Sep,2004)

8. Java SE 6 (11th Dec,2006)

9. Java SE 7 (28th July, 2011)

10.Java SE 8 (18th March,2014)

**Features of Java**

There is given many features of java. They are also known as java buzzwords. The Java Features given below are simple and easy to understand.

1. Simple

2. Object-Oriented

3. Portable

4. Platform independent

5. Secured

6. Robust

7. Architecture neutral

8. Dynamic

9. Interpreted

10. High Performance

11. Multithreaded

12.Distributed

**Java Comments**

The java comments are statements that are not executed by the compiler and interpreter. The comments can be used to provide information or explanation about the variable, method, class or any statement. It can also be used to hide program code for specific time.

Types of Java Comments There are 3 types of comments in java.

1. Single LineComment

2. Multi LineComment

3. Documentation Comment Java Single Line Comment The single line comment is used to comment only one line.

Syntax:

1. //This is single line comment

Example:

public class CommentExample1

{

public static void main(String[] args) {

int i=10;//Here, i is a variable System.out.println(i);

}

}

###### Class Names and Identifiers

By convention, class names begin with a capital letter and capitalize the first letter of each word they include (e.g., SampleClassName). A class name is an **identifier**—a series of characters consisting of letters, digits, underscores (\_) and dollar signs ($) that does

not begin with a digit and does not contain spaces. Some valid identifiers are Welcome1,

$value, \_value, m\_inputField1 and button7. The name 7button is not a valid identifier because it begins with a digit, and the name input field is not a valid identifier because it contains a space. Normally, an identifier that does not begin with a capital letter is not a class name. Java is **case sensitive**—uppercase and lowercase letters are distinct—so value and Value are different (but both valid) identifiers.

A **left brace** (as in line 5), **{**, begins the **body** of every class declaration. A corresponding

**right brace** (at line 11), **}**, must end each class declaration. Lines 6–10 are indented.

###### Declaring a Method

Line 6

##### // main method begins execution of Java application

is an end-of-line comment indicating the purpose of lines 7–10 of the program. Line 7

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

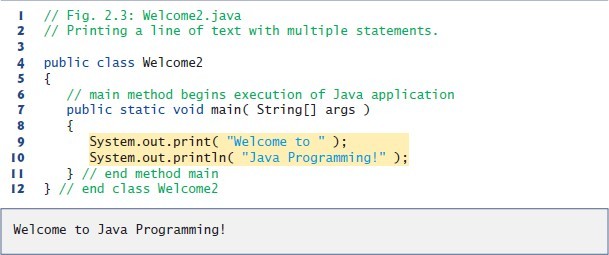
is the starting point of every Java application. The **parentheses** after the identifier main indicate that it’s a program building block called a **method**. Java class declarations normally contain one or more methods. For a Java application, one of the methods *must* be called main and must be defined as shown in line 7; otherwise, the Java Virtual Machine (JVM) will not execute the application. Methods perform tasks and can return information when they complete their tasks. Keyword **void** indicates that this method will *not* return any information. Later, we’ll see how a method can return information. For now, simply mimic main’s first line in your Java applications. In line 7, the String[] args in parentheses is a required part of the method main’s declaration. The left brace in line 8 begins the **body of the method declaration**. A corresponding right brace must end it (line 10). Line 9 in the method body is indented between the braces.

##### Modifying Your First Java Program

In this section, we modify the example in Fig. 2.1 to print text on one line by using multiple statements and to print text on several lines by using a single statement.

###### Displaying a Single Line of Text with Multiple Statements

Welcome to Java Programming! can be displayed several ways. Class Welcome2, shown in Fig., uses two statements (lines 9–10) to produce the output.



The program is similar to previous program, so we discuss only the changes here. Line 2

##### // Printing a line of text with multiple statements.

is an end-of-line comment stating the purpose of the program. Line 4 begins the Welcome2 class declaration. Lines 9–10 of method main

##### System.out.print( "Welcome to " );

##### System.out.println(“Java Programming”);

**Conclusion:**

I can learn the basic concepts of java & learn how to execute any program in java also learn the basic syntax of java.

**Completion Date: Co-Ordinator Sign:**