**B1. Features of Java. (or) Java buzzwords.**

**:- Objective:**

Simple, Secure, Portable, Object-oriented, Robust, Multithreaded, Architecture-neutral, In-terpreted, High Performance, Distributed and Dynamic are the Java buzzwords. This section go in details in each buzzwords and explain about these terms.

**Overview:**

Two main feature of Java is Portable and Secure but there other reasons why Java is consid-ered as a powerful programming language are discussed below. These buzzwords were also discussed by the Java team to describe the power of Java .

•Simple

• Secure

• Portable

• ObjectOriented

• Robust

• Multithreaded

• Architecture Neutral

• Interpreted

• High Performance

• Distributed

• Dynamic

**Simple**: Java inherits all the best features from the programming languages like C, C++ and thus makes it really easy for any developer to learn with little programming experience. The concept of Object Oriented programming was not invented by Java but it was just adopted by the Java team. Programming notation of Java is not different from the program-ming language like C and C++ which makes developers have little trouble to learn Java.

**Secure:** When Java programs are executed they don’t instruct commands to the machine directly. Instead Java Virtual machine reads the program (ByteCode) and convert it into the machine instructions. This way any program tries to get illegal access to the system will not be allowed by the JVM. Allowing Java programs to be executed by the JVM makes Java program fully secured under the control of the JVM.

**Portable:**Java programs are portable because of its ability to run the program on any plat-form and no dependency on the underlying hardware / operating system.

Java programs are portable because of its ability to run the program on any plat-form and no dependency on the underlying hardware / operating system.

**Object Oriented Programming Language**: Java programming language was influenced from its previous successors programming language like C++. Java developers did not just took everything and implemented in Java but they analysed the current challenges in the existing language and then included what is necessary. Java is object-oriented programming language but everything in Java are not objects. Java manages to maintain balance and adopted what make sense in the current situation. The object oriented model in Java is simple and easy to extend and also the primitive types such as integers, are retained for high-performance.

**Robust:** Following features of Java make it Robust.

•Platform Independent

•Object Oriented Programming Language

•Memory management

•Exception HandlingPlatform Independent

:Java program are written once and executed on any platform this makes the job of developer easier to develop programs and not code machine dependent coding.

**Object Oriented Programming Language**: Helps to break the complex code into easy to understand objects and manage high complexity programs in distributed team environment.

**Memory Management**: In traditional programming language like C, C++ user has to man-age memory by allocating and deallocating memory which leads to memory leaks in the program. In Java, memory management is taken care by the Java Virtual Machine and safe from memory crashes. All the allocation and clean of the memory is done automatically.

**Exception Handling**: In Java, developers are forced to handle the exception during the development time. All the possible exception are errored out during the compilation of the program. This way when the exception happens during runtime there is proper exception handling mechanism already coded in the program

**Multithreaded:** Java allows you to develop program that can do multiple task simultaneously. Interactive based programming allows you to write program that responds to the user actions and helps developers to just implement the logic based on the user action instead to manage the complete multitasking solution.

**Architecture-Neutral:** The major challenge when Java was developing is to have programming language with which a program can be developed and executed anytime in future. With changing environments of hardware, processor, Operating system there was need to have program still adopt to this architecture changes. Java code does not depend on the underlying architecture and only depends on it JVM thus accomplish the architecture neutral programming language.

**Interpreter:** The compiled code of Java is not machine instructions but rather it’s intermediate code called Byte Code. This code can be executed on any machine that implements the Java virtual Machine. JVM interprets the ByteCode into Machine instructions during runtime.

**High Performance:** When java programs are executed, JVM does not interpret entire code into machine instructions. If JVM attempts to do this then there will huge performance im-pact for the high complexity programs. JVM was intelligently developed to interpret only the piece of the code that is required to execute and untouched the rest of the code. The performance of java is never questioned compared with another programming language.

**Distributed:** Java has a feature called Remote Method Invocation (RMI) using which a program can invoke method of another program across a network and get the output.

**Dynamic:** Java programs access various runtime libraries and information inside the com-piled code (Bytecode). This dynamic feature allows to update the pieces of libraries without affecting the code using it.