Java Fundamental

CLASS- 1

Introduction & History

Anshuman Dixit

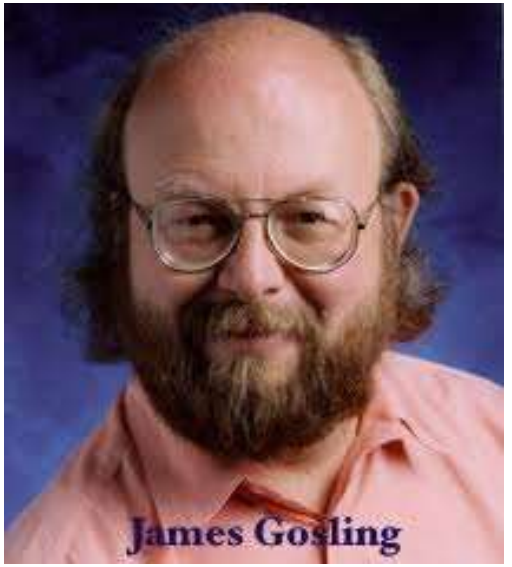
Introduction

- Java is a high level, object oriented and secure programming language.
- Java is general purpose, platform independent programming language.
- Due to robustness and high security, Java is used as mostly used and comfortable programming languages as per user's need.

History of Java

Java was developed by ***Sun Microsystems*** (which is now the subsidiary of Oracle) in the year 1995.

James Gosling is known as the father of Java.



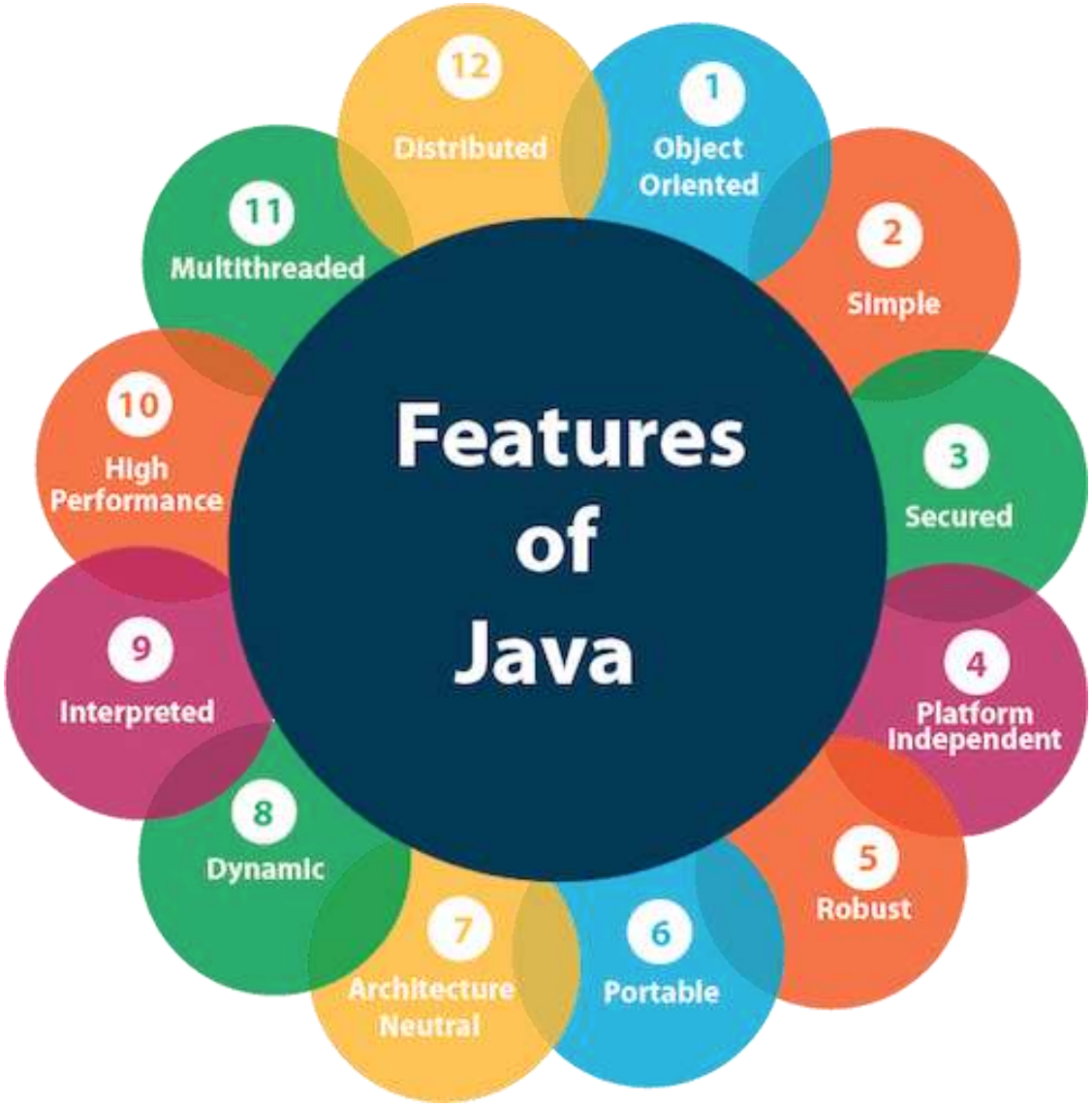
History of Java

- Java was originally designed for interactive television, but it was too advanced technology for the digital cable television industry at the time.
- The principles for creating Java programming were "Simple, Robust, Portable, Platform-independent, Secured, High Performance, Multithreaded, Architecture Neutral, Object-Oriented, Interpreted, and Dynamic".
- In 1995, Time magazine called **Java one of the Ten Best Products of 1995**.

History of Java versions

- JDK Alpha and Beta (1995)
- JDK 1.0 (23rd Jan 1996)
- JDK 1.1 (19th Feb 1997)
- J2SE 1.2 (8th Dec 1998)
- J2SE 1.3 (8th May 2000)
- J2SE 1.4 (6th Feb 2002)
- J2SE 5.0 (30th Sep 2004)
- Java SE 6 (11th Dec 2006)
- Java SE 7 (28th July 2011)
- Java SE 8 (18th Mar 2014)
- Java SE 9 (21st Sep 2017)
- Java SE 10 (20th Mar 2018)

Features of Java



Features of Java

- Simple
- Object-Oriented
- Portable
- Platform independent
- Secured
- Robust
- Architecture neutral

Features of Java

- Easy to understand coding approach
- Interpreted
- High Performance
- Multithreaded
- Distributed
- Dynamic
- Large scale of packages.

C++ vs Java

C++	Java
Platform dependent	Platform Independent
Mainly used for system programming	Used for window, web-based, enterprise and mobile applications
C++ was designed for systems and applications programming. It was an extension of C	It was designed with a goal of being easy to use and accessible to a broader audience.
Support goto statement	Doesn't support
Support multiple inheritance	Doesn't support multiple inheritance*
Support pointers	doesn't support pointers
Support operator overloading	Doesn'Support operator overloading
supports both call by value and call by reference.	Support only call by value
Contains structure and union	No structure and union

C++ vs Java

C++	Java
C++ uses compiler only so it is machine dependent	It uses both compiler and interpreted, so machine independent
No threads	Contains threads
C++ supports virtual keyword so that we can decide whether or not override a function	Java has no virtual keyword. We can override all non-static methods by default. In other words, non-static methods are virtual by default.
creates a new inheritance tree always.	Java uses a single inheritance tree always because all classes are the child of Object class in java.
C++ is nearer to hardware.	Java is not so interactive with hardware.
Contains header files	Do not contains header files
Called as partial object oriented programming language	Called as pure object oriented programming language

- Object Oriented Concepts
- Java installation
- Java components