ECAP615

Programming in Java

Harjinder Kaur

Assistant Professor

Learning Outcomes

After this lecture, you will be able to

- understand the importance of Java programing
- learn the comparative analysis between java and C++
- identify the various features of Java
- know the different operators and data types of java

What is JAVA?

- Java is a programming language and a platform.

What is JAVA?

- Java is a programming language and a platform.
- Java is a high level, robust, object-oriented and secure programming language.

What is JAVA?

- Java is a programming language and a platform.
- Java is a high level, robust, object-oriented and secure programming language.
- Java was developed by Sun Microsystems in the year 1995.

What is JAVA?

- James Gosling is known as the father of Java.

What is JAVA?

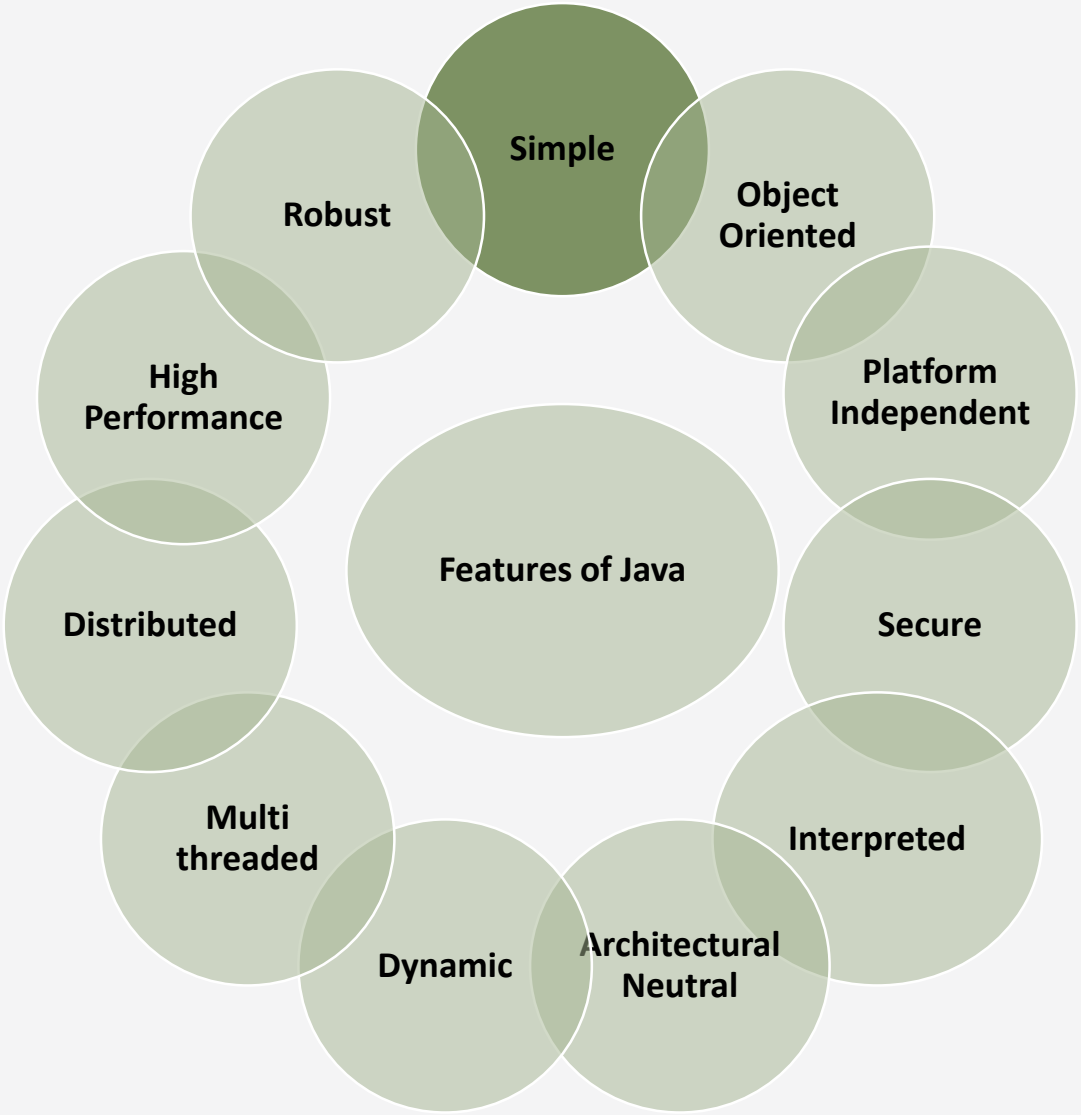
- James Gosling is known as the father of Java.
- Before Java, its name was Oak.

Java Platforms / Editions

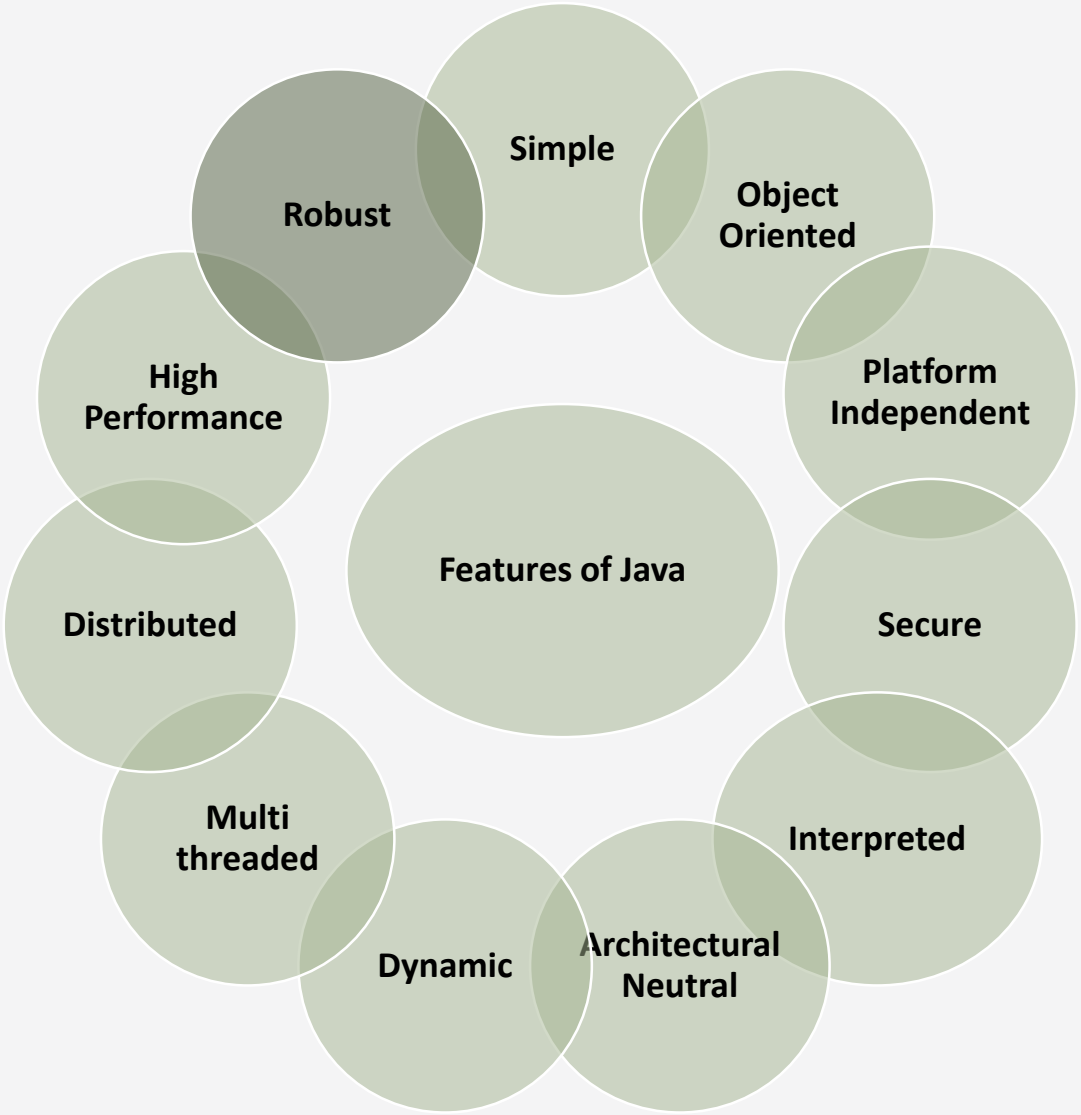
There are 4 platforms or editions of Java:

1. Java SE (Java Standard Edition)
2. Java EE (Java Enterprise Edition)
3. Java ME (Java Micro Edition)
4. JavaFX

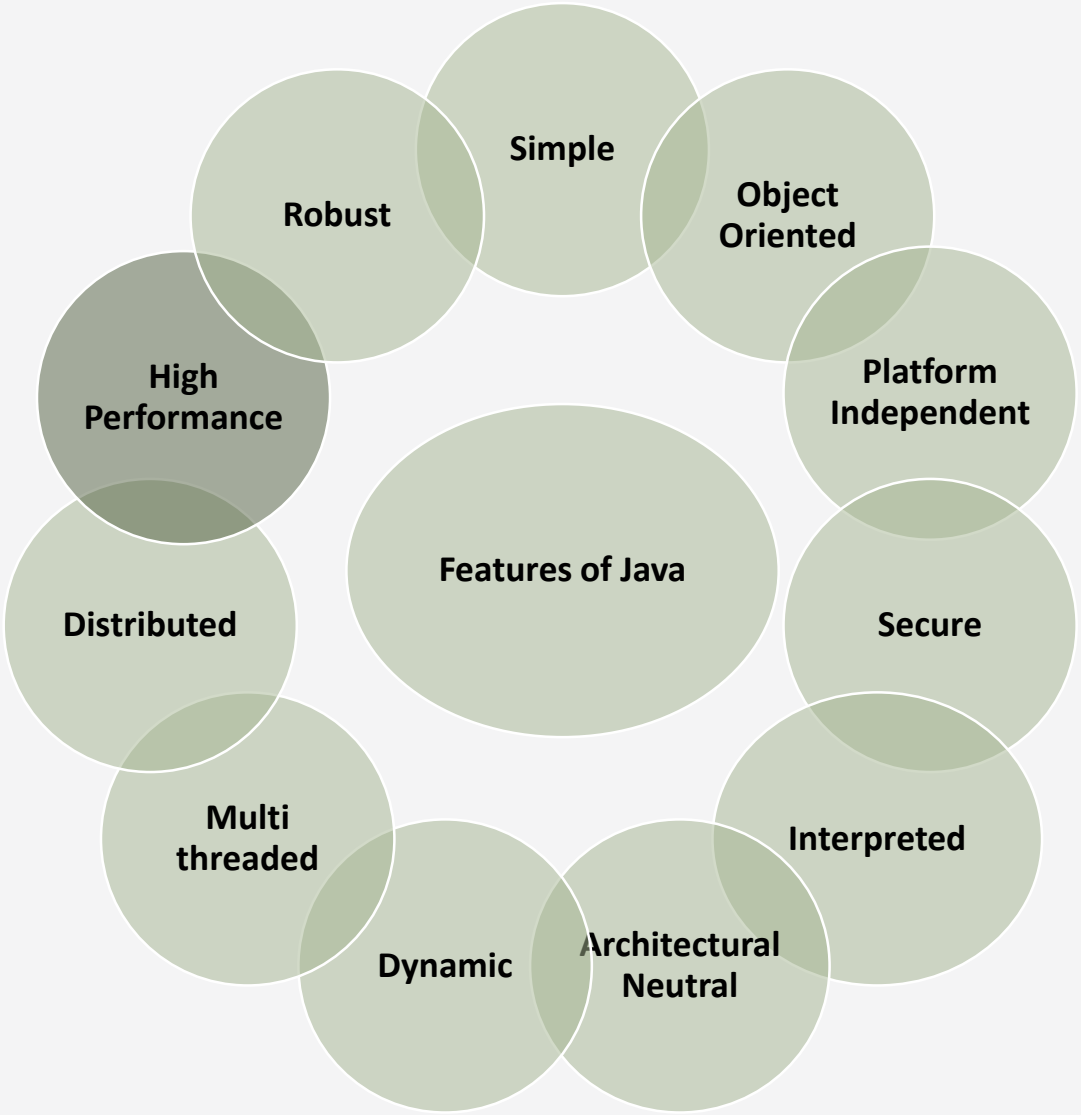
Features of Java



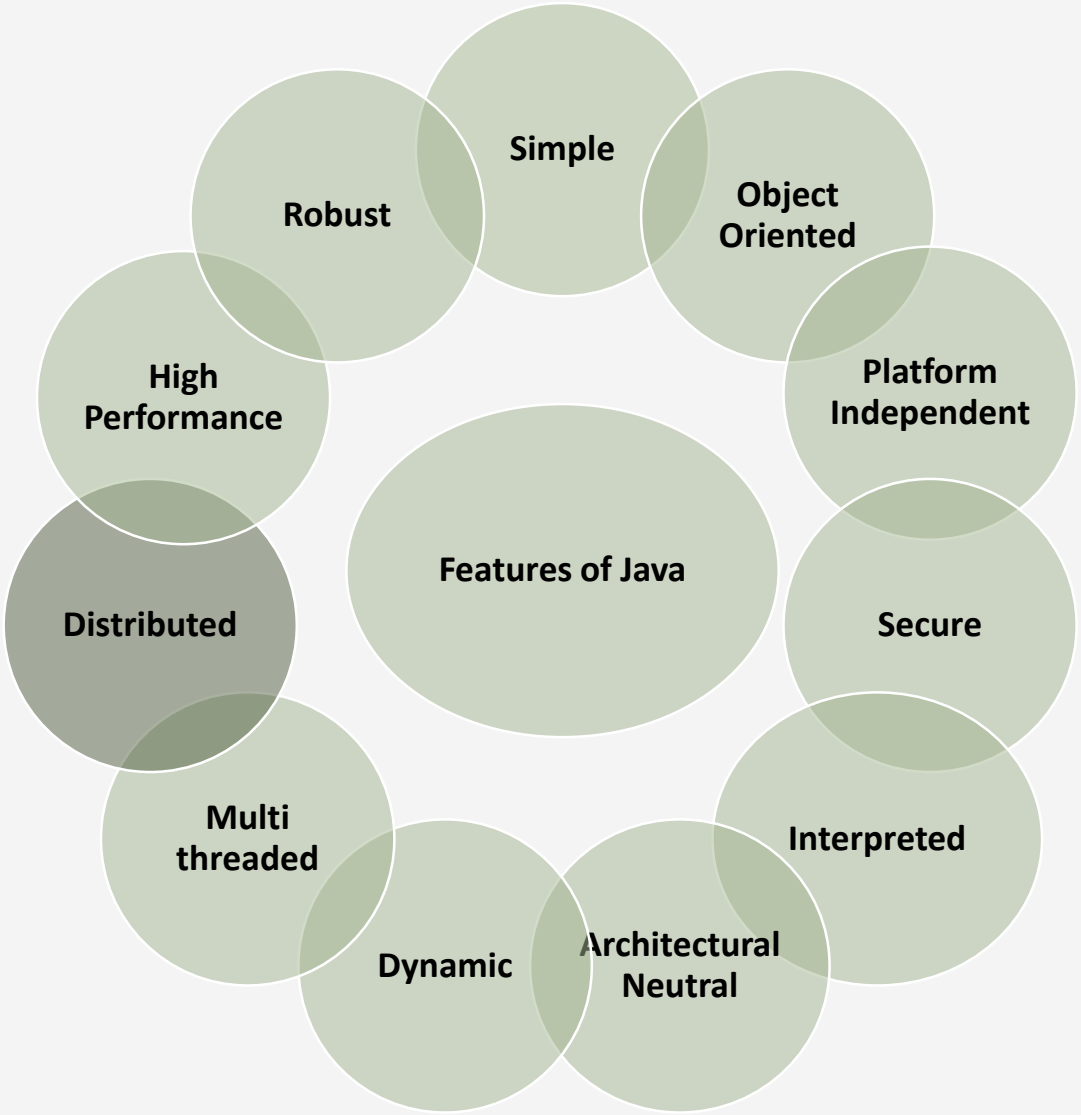
Features of Java



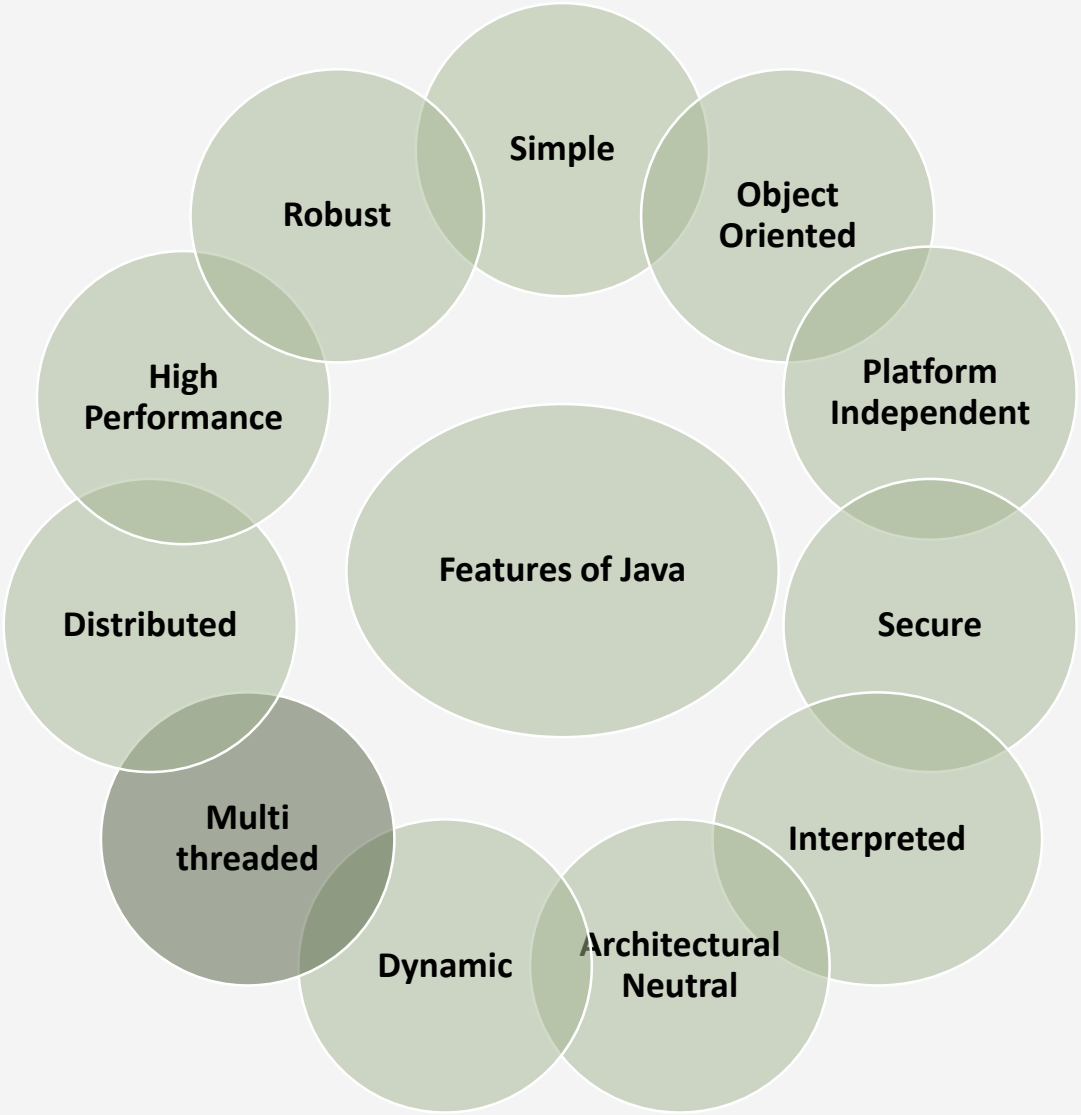
Features of Java



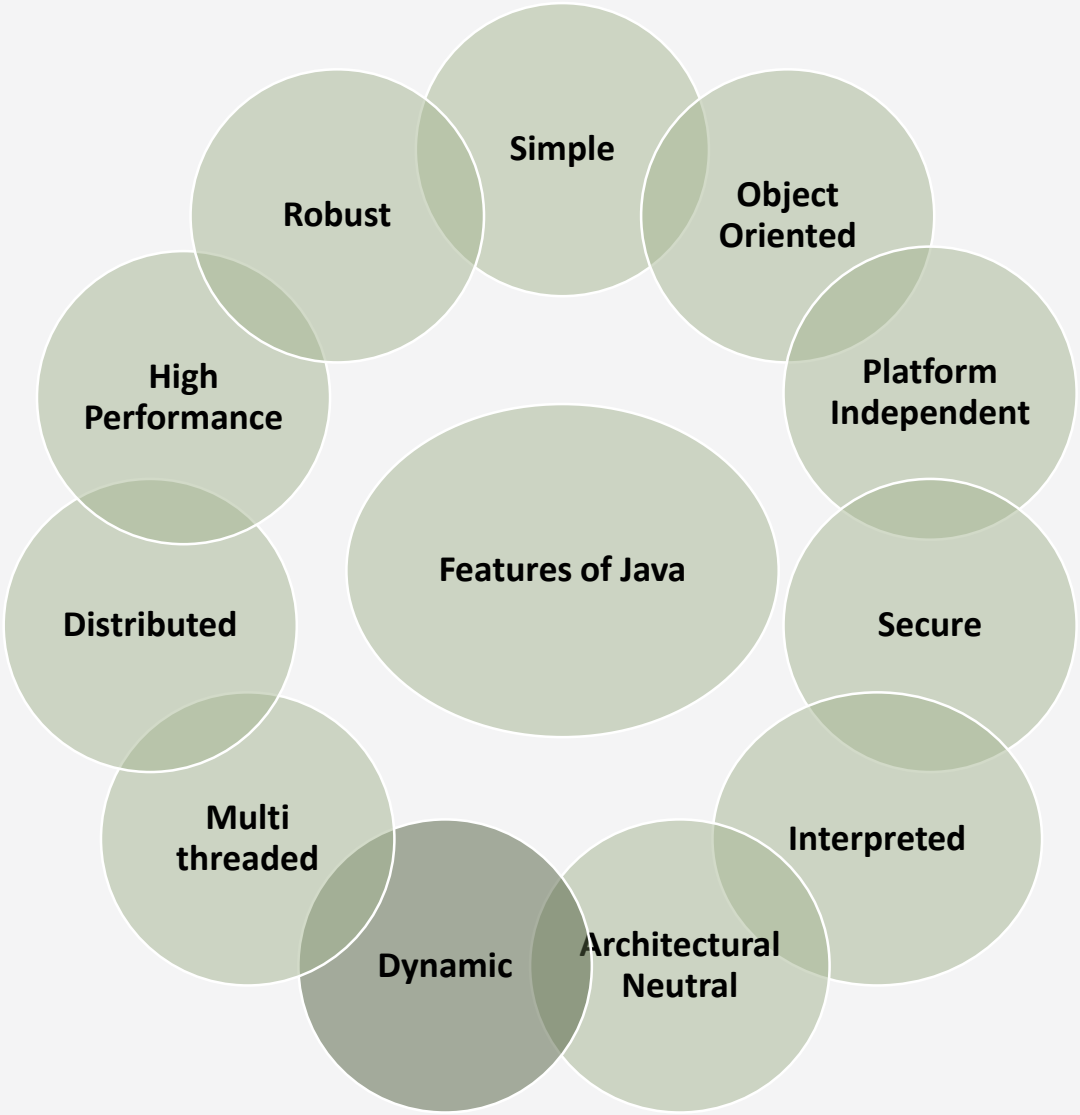
Features of Java



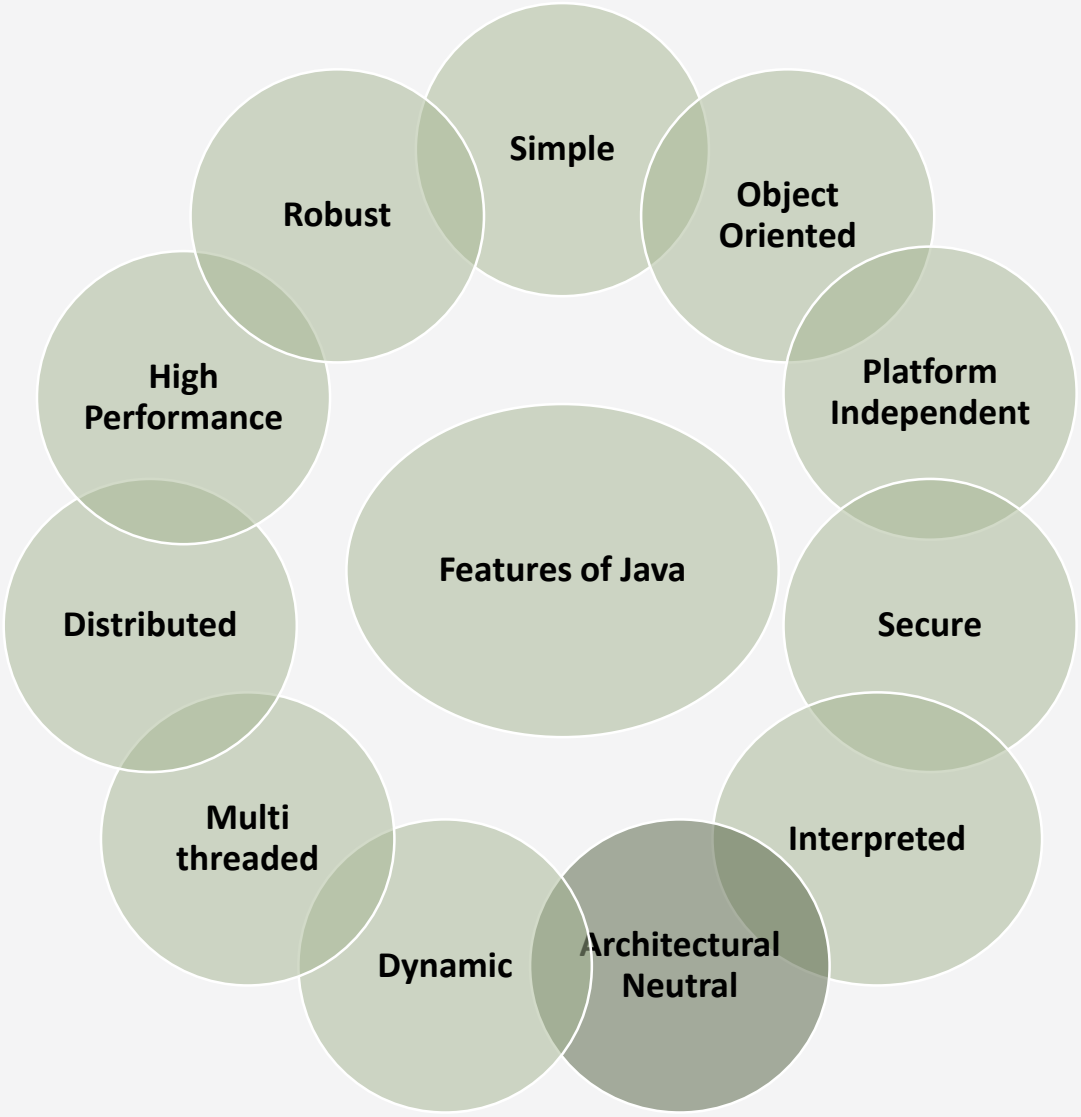
Features of Java



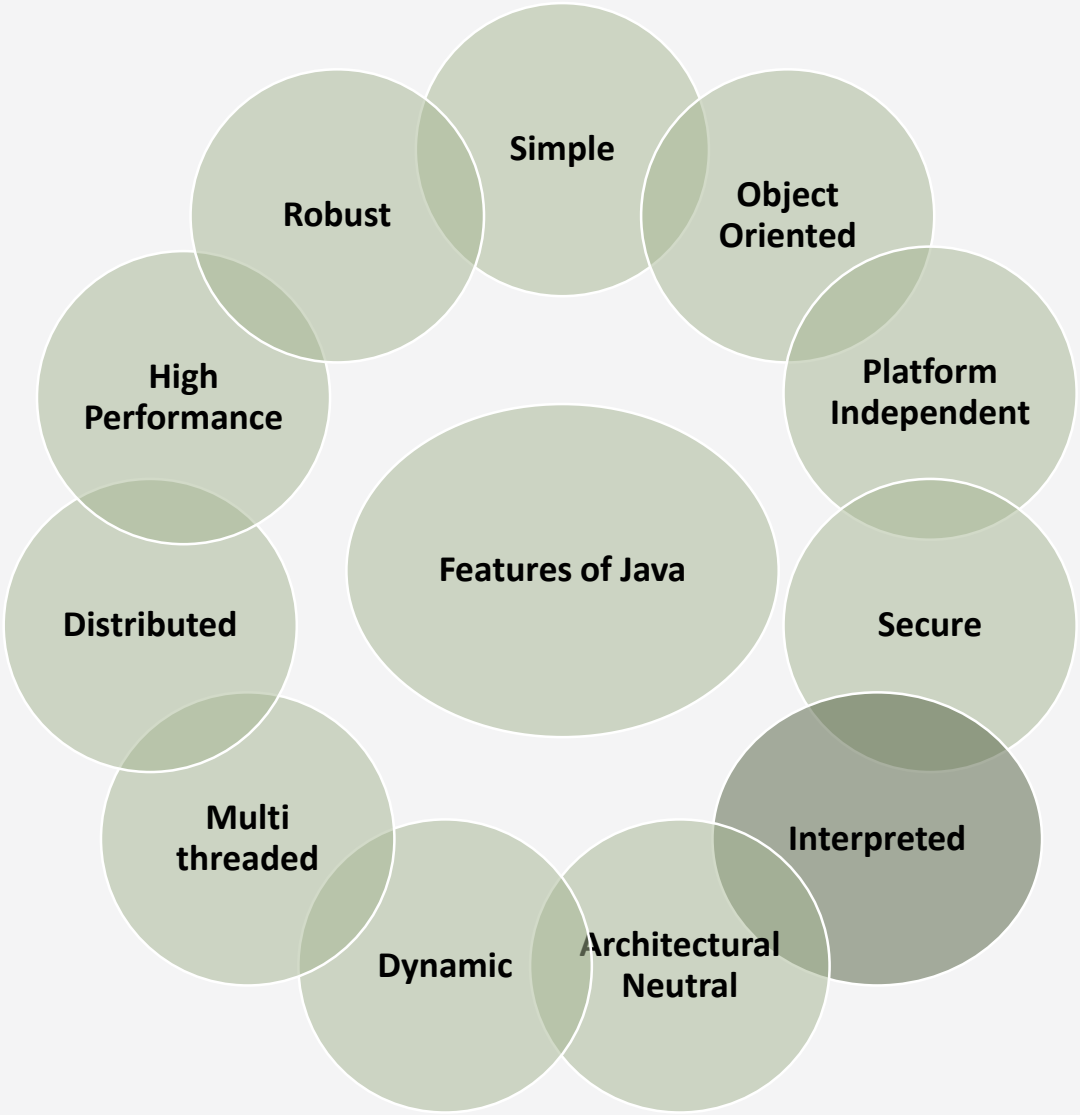
Features of Java



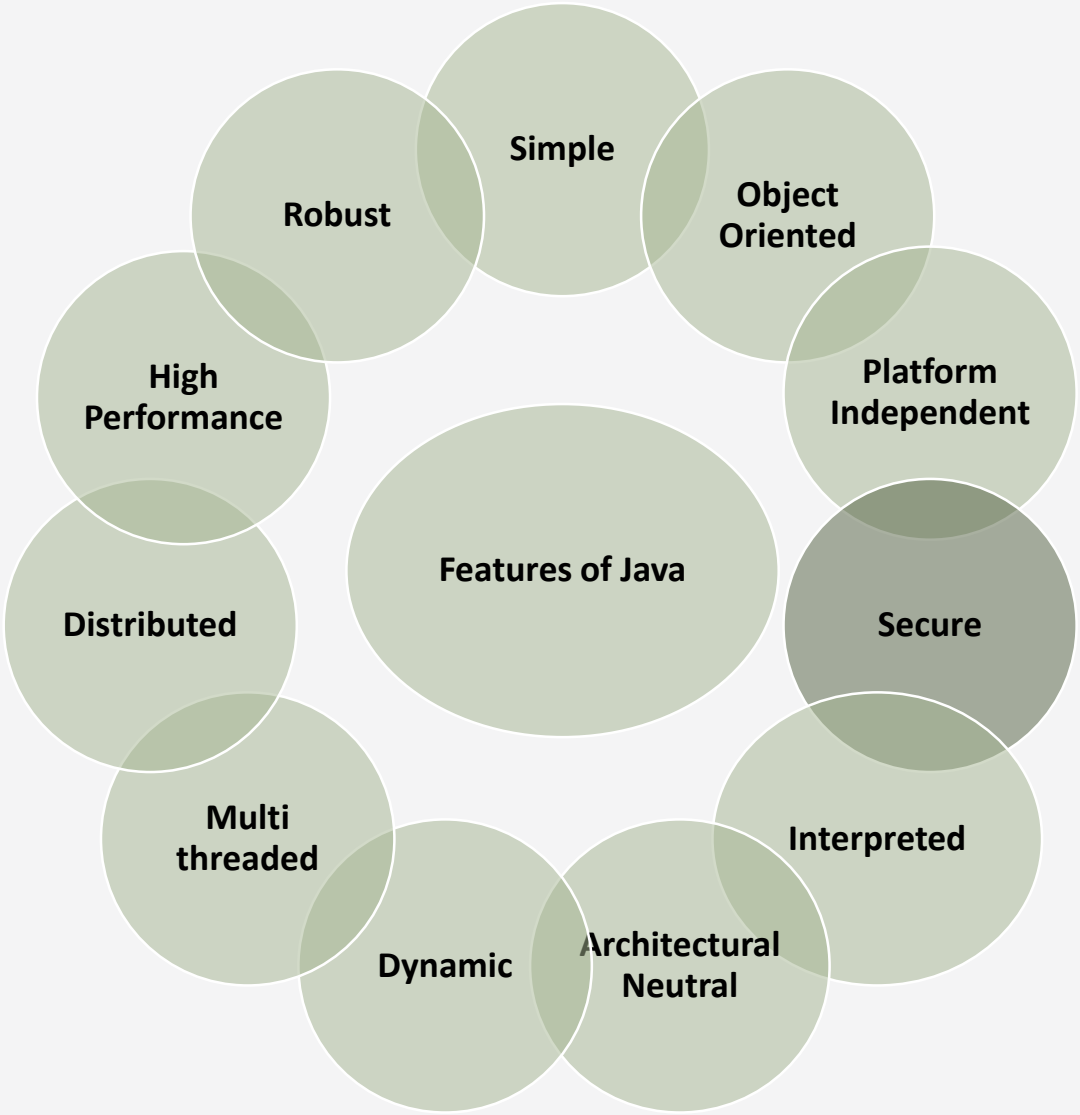
Features of Java



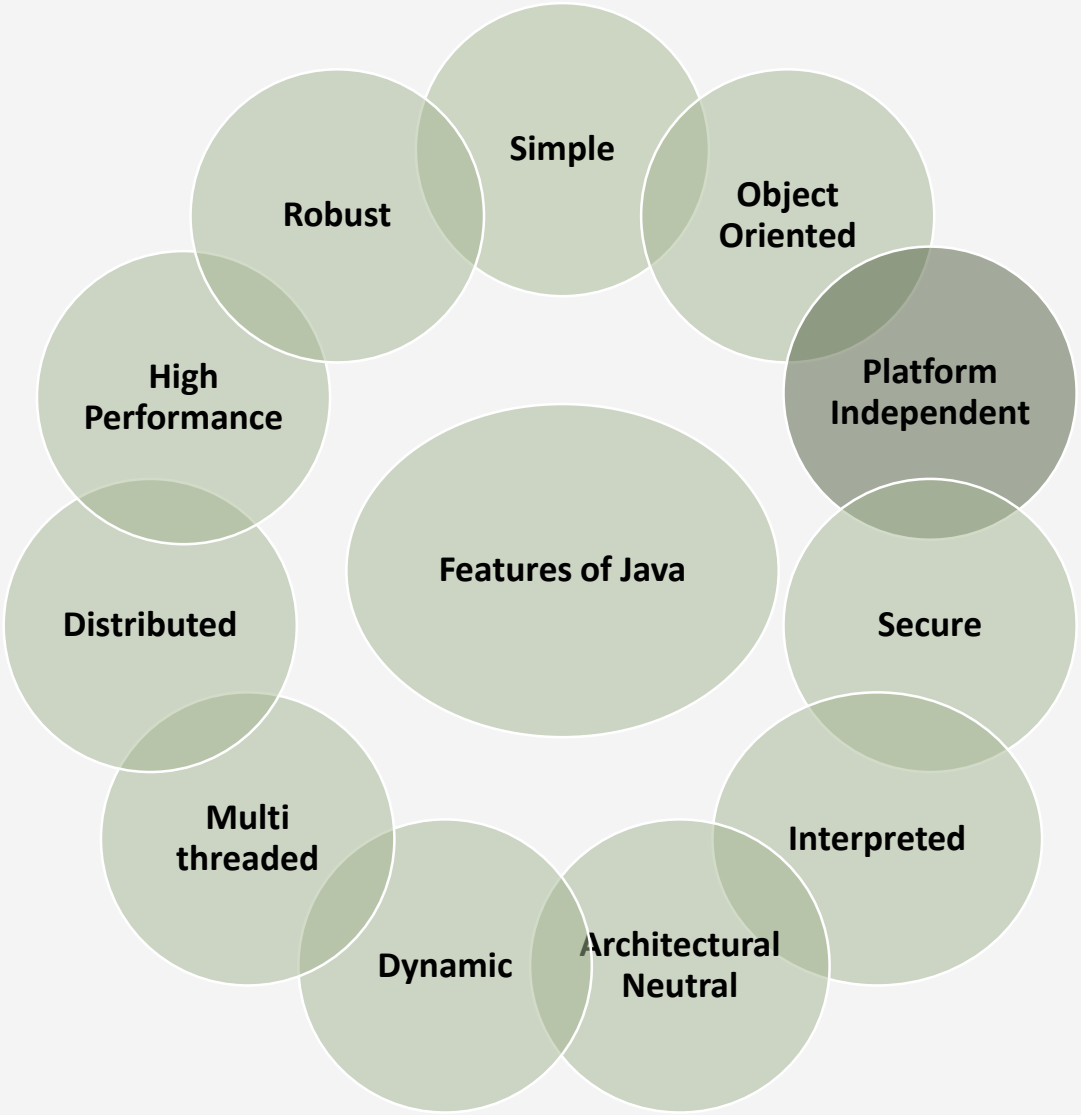
Features of Java



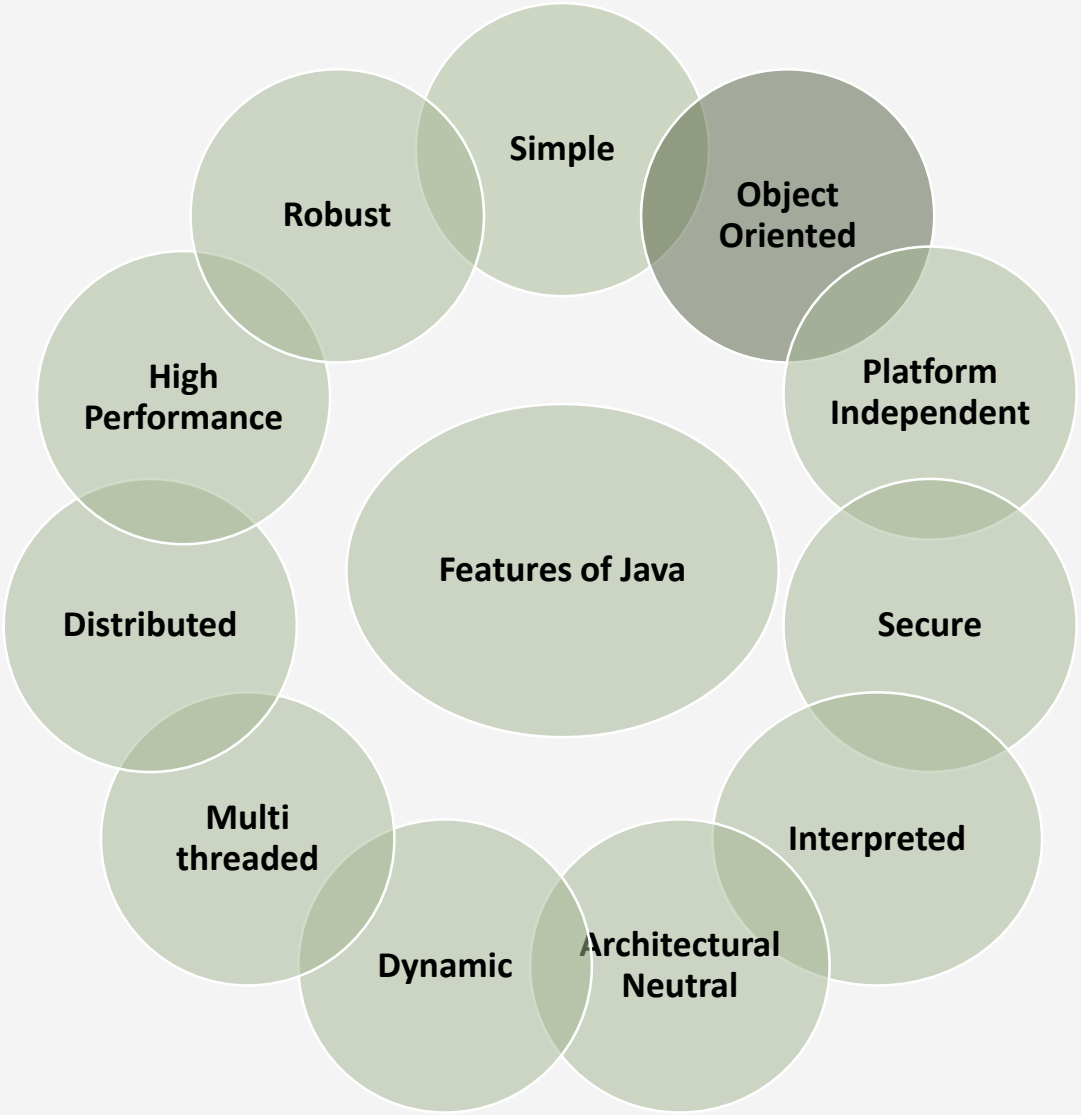
Features of Java



Features of Java



Features of Java



Comparison between C++ and Java

Features	C++	Java
Platform-independent	C++ is platform-dependent.	Java is platform-independent.
Mainly used for	C++ is mainly used for system programming.	Java is mainly used for application programming.
Goto	C++ supports the go to statement.	Java doesn't support the goto statement.
Multiple inheritance	C++ supports multiple inheritance.	Java doesn't support multiple inheritance. It can be achieved by interfaces in java.
Operator Overloading	C++ supports operator overloading.	Java doesn't support operator overloading.
Pointers	C++ supports pointers. You can write pointer program in C++.	You can't write the pointer program in java.

Comparison between C++ and Java

Features	C++	Java
Compiler and Interpreter	C++ uses compiler only.	Java uses compiler and interpreter both.
Call by Value and Call by reference	C++ supports both call by value and call by reference.	Java supports call by value only. There is no call by reference in java.
Structure and Union	C++ supports structures and unions.	Java doesn't support structures and unions.
Thread Support	C++ doesn't have built-in support for threads. It relies on third-party libraries for thread support.	Java has built-in thread support.
Hardware	C++ is nearer to hardware.	Java is not so interactive with hardware.

Comparison between C++ and Java

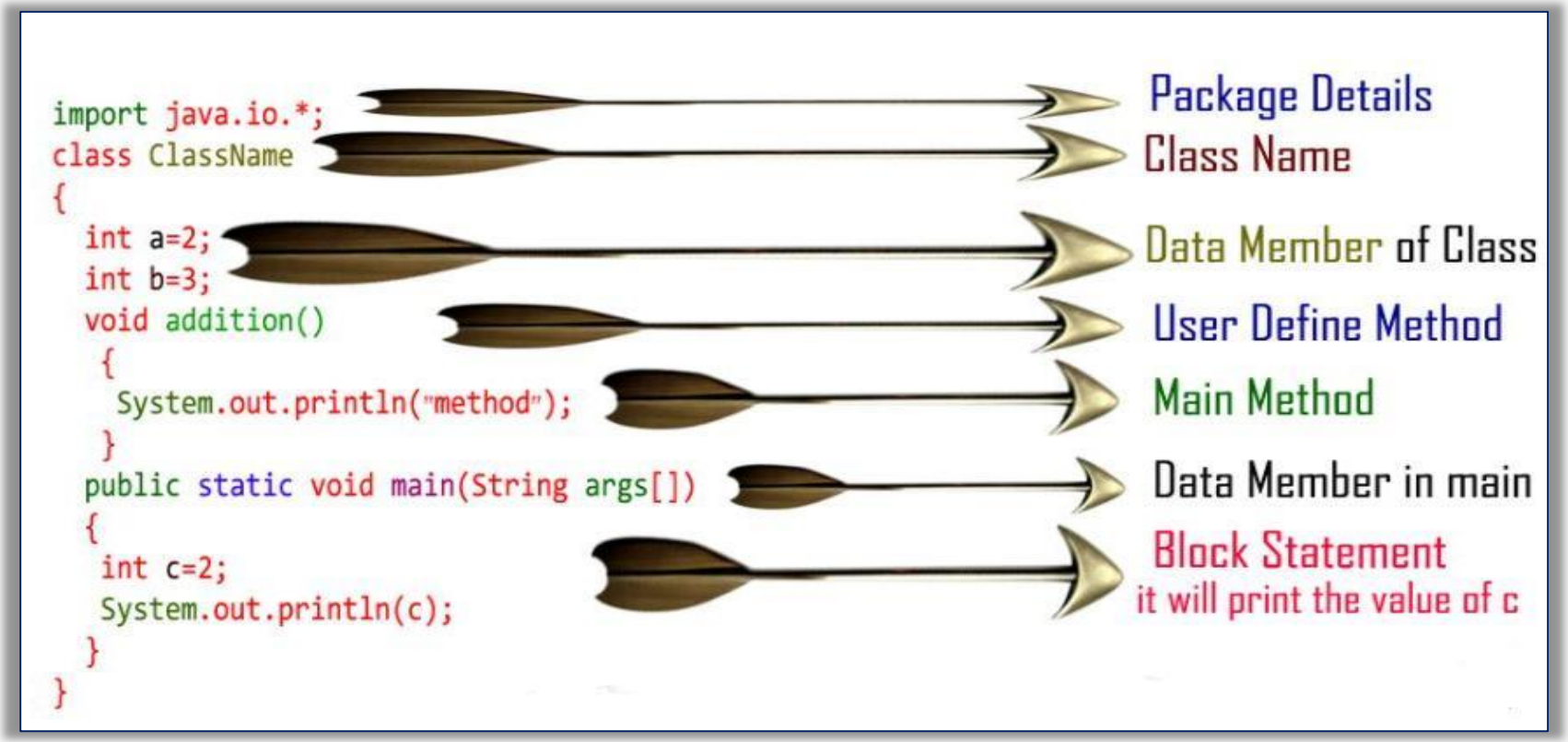
Features	C++	Java
Documentation comment	C++ doesn't support documentation comment.	Java supports documentation comment (/** ... */) to create documentation for java source code.
Virtual Keyword	C++ supports virtual keyword so that we can decide whether or not override a function.	Java has no virtual keyword.Non-static methods are virtual by default.
Inheritance Tree	C++ creates a new inheritance tree always.	Java uses a single inheritance tree always because all classes are the child of Object class in java. The object class is the root of the inheritance tree in java.

Structure of Java Program

Documentation Section	Suggested
Package Statement	Optional
Import Statement	Optional
Interface Statement	Optional
Class Definitions	Optional

Main Method Class
{
Main method Definition
}
	Essential

Sample Program



Operators in Java

Operator in Java is a symbol which is used to perform operations. For example: +, -, *, / etc. There are many types of operators in Java which are given below:

- Unary Operator
- Arithmetic Operator
- Shift Operator
- Relational Operator
- Bitwise Operator
- Logical Operator
- Ternary Operator
- Assignment Operator

Operators & their Precedence

Operator Type	Category	Precedence
Unary	postfix	expr++ expr--
	prefix	++expr --expr
Arithmetic	multiplicative	* / %
	additive	+ -
Shift	shift	<< >> >>>
Relational	comparison	< > <= >= instanceof
	equality	== !=
Bitwise	bitwise AND	&
	bitwise exclusive OR	^
	bitwise inclusive OR	|

Operators & their Precedence

Operator Type	Category	Precedence
Logical	logical AND	&&
	logical OR	||
Ternary	ternary	? :
Assignment	assignment	= += -= *= /= %= &= ^= |= <<= >>= >>>=

Data Types

Data types specify the different sizes and values that can be stored in the variable. There are two types of data types in Java:

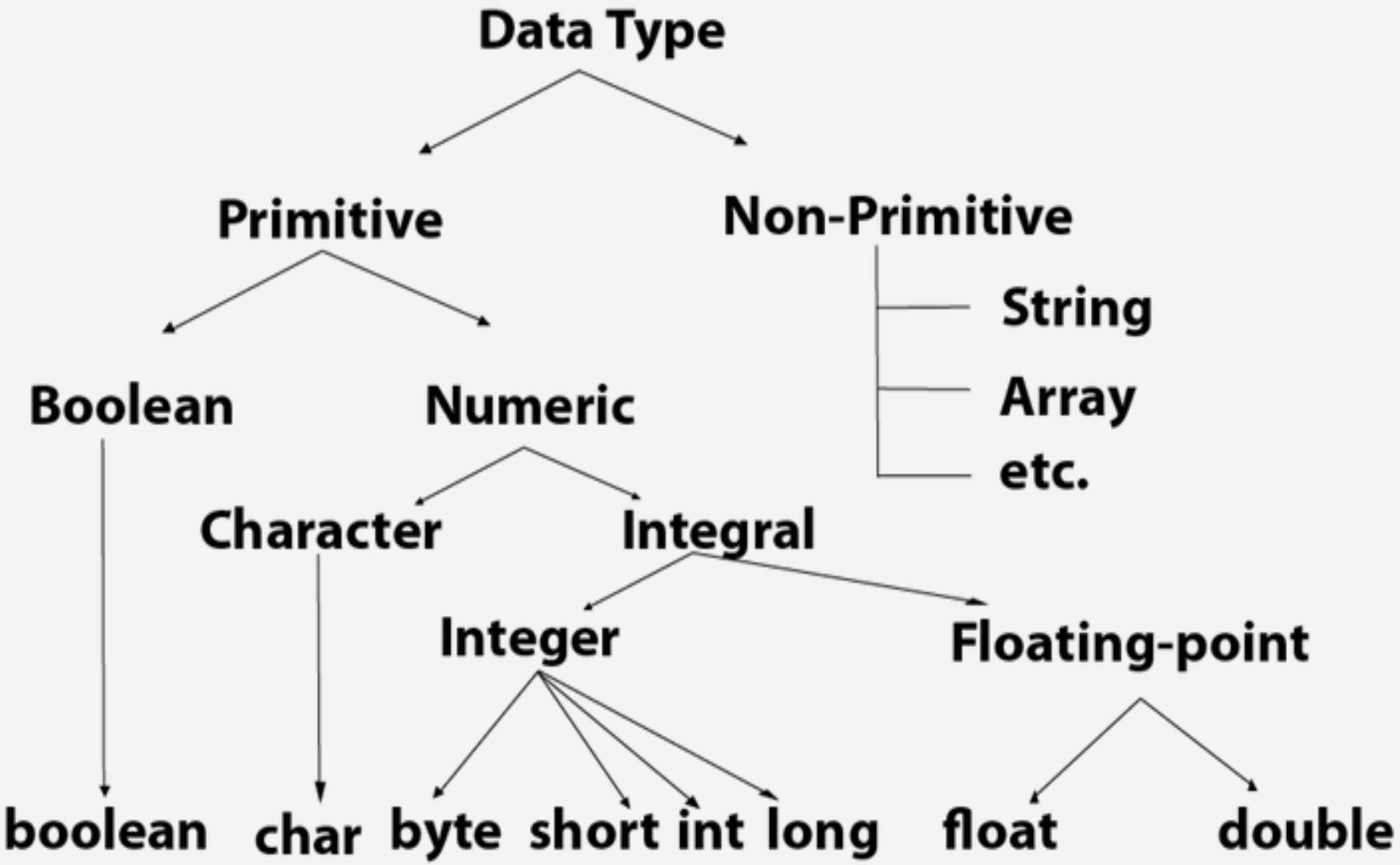
- Primitive data types: The primitive data types include boolean, char, byte, short, int, long, float and double.
- Non-primitive data types: The non-primitive data types include Classes, Interfaces, and Arrays.

Data Types

Data types specify the different sizes and values that can be stored in the variable. There are two types of data types in Java:

- Primitive data types: The primitive data types include boolean, char, byte, short, int, long, float and double.
- Non-primitive data types: The non-primitive data types include Classes, Interfaces, and Arrays.

Data Types


Data Type
Primitive
Non-Primitive
String
Array
etc.
Boolean
Numeric
Character
Integral
Integer
Floating-point
boolean
char
byte
short
int
long
float
double

Data Type with default values

Data Type	Default Value	Default size
boolean	false	1 bit
char	'\u0000'	2 byte
byte	0	1 byte
short	0	2 byte
int	0	4 byte
long	0L	8 byte
float	0.0f	4 byte
double	0.0d	8 byte

Boolean Data Type

The Boolean data type is used to store only two possible values:

- true
- false.

Example: Boolean one = false

Byte Data Type

- It is an 8-bit signed two's complement integer.
- Its value-range lies between -128 to 127 (inclusive).
- Its minimum value is -128 and maximum value is 127. Its default value is 0.
- It saves space because a byte is 4 times smaller than an integer.

Example: byte a = 10, byte b = -20

Byte Data Type

- It is an 8-bit signed two's complement integer.
- Its value-range lies between -128 to 127 (inclusive).
- Its minimum value is -128 and maximum value is 127. Its default value is 0.
- It saves space because a byte is 4 times smaller than an integer.

Example: byte a = 10, byte b = -20

Byte Data Type

- It is an 8-bit signed two's complement integer.
- Its value-range lies between -128 to 127 (inclusive).
- Its minimum value is -128 and maximum value is 127. Its default value is 0.
- It saves space because a byte is 4 times smaller than an integer.

Example: byte a = 10, byte b = -20

Byte Data Type

- It is an 8-bit signed two's complement integer.
- Its value-range lies between -128 to 127 (inclusive).
- Its minimum value is -128 and maximum value is 127. Its default value is 0.
- It saves space because a byte is 4 times smaller than an integer.

Example: byte a = 10, byte b = -20

Short Data Type

- The short data type is a 16-bit signed two's complement integer.
- Its value-range lies between -32,768 to 32,767 (inclusive).
- A short data type is 2 times smaller than an integer.

Example: short s = 10000, short r = -5000

Short Data Type

- The short data type is a 16-bit signed two's complement integer.
- Its value-range lies between -32,768 to 32,767 (inclusive).
- A short data type is 2 times smaller than an integer.

Example: short s = 10000, short r = -5000

Short Data Type

- The short data type is a 16-bit signed two's complement integer.
- Its value-range lies between -32,768 to 32,767 (inclusive).
- A short data type is 2 times smaller than an integer.

Example: short s = 10000, short r = -5000

int data type

- The int data type is a 32-bit signed two's complement integer.
- Its value-range lies between - 2,147,483,648 (-2^31) to 2,147,483,647 (2^31 -1) (inclusive).

Example: int a = 100000, int b = -200000

int data type

- The int data type is a 32-bit signed two's complement integer.
- Its value-range lies between - 2,147,483,648 (-2^31) to 2,147,483,647 (2^31 -1) (inclusive).

Example: int a = 100000, int b = -200000

Float Data Type

- The float data type is a single-precision 32-bit IEEE 754 floating point.
- Its value range is unlimited.
- The float data type should never be used for precise values, such as currency.
- Its default value is 0.0F.

Example: float f1 = 234.5f

Float Data Type

- The float data type is a single-precision 32-bit IEEE 754 floating point.
- Its value range is unlimited.
- The float data type should never be used for precise values, such as currency.
- Its default value is 0.0F.

Example: float f1 = 234.5f

Float Data Type

- The float data type is a single-precision 32-bit IEEE 754 floating point.
- Its value range is unlimited.
- The float data type should never be used for precise values, such as currency.
- Its default value is 0.0F.

Example: float f1 = 234.5f

Float Data Type

- The float data type is a single-precision 32-bit IEEE 754 floating point.
- Its value range is unlimited.
- The float data type should never be used for precise values, such as currency.
- Its default value is 0.0F.

Example: float f1 = 234.5f

Double Data Type

- The double data type is a double-precision 64-bit IEEE 754 floating point.
- Its value range is unlimited.
- The double data type is generally used for decimal values just like float.

Example: double d1 = 12.3

Double Data Type

- The double data type is a double-precision 64-bit IEEE 754 floating point.
- Its value range is unlimited.
- The double data type is generally used for decimal values just like float.

Example: double d1 = 12.3

Double Data Type

- The double data type is a double-precision 64-bit IEEE 754 floating point.
- Its value range is unlimited.
- The double data type is generally used for decimal values just like float.

Example: double d1 = 12.3

Char Data Type

- The char data type is a single 16-bit Unicode character.
- Its value-range lies between '\u0000' (or 0) to '\uffff' (or 65,535 inclusive).
- The char data type is used to store characters.

Example: char letterA = 'A'

Char Data Type

- The char data type is a single 16-bit Unicode character.
- Its value-range lies between '\u0000' (or 0) to '\uffff' (or 65,535 inclusive).
- The char data type is used to store characters.

Example: char letterA = 'A'

Char Data Type

- The char data type is a single 16-bit Unicode character.
- Its value-range lies between '\u0000' (or 0) to '\uffff' (or 65,535 inclusive).
- The char data type is used to store characters.

Example: char letterA = 'A'


That’s all for now…