



SNS COLLEGE OF ENGINEERING



Kurumbapalayam(Po), Coimbatore - 641 107

Accredited by NAAC-UGC with 'A' Grade

Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

Department of Information Technology

19CS204 OBJECT ORIENTED PROGRAMMING

I YEAR /II SEMESTER

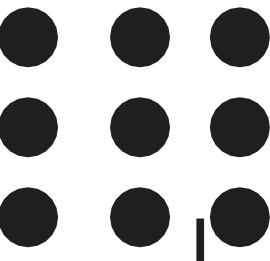
Unit 1- INTRODUCTION TO OOP

JRE and Data Types, Operators





JVM



Java Virtual machine (JVM) is the virtual machine that runs the Java bytecodes.

- Mostly in other Programming Languages, compiler produce code for a particular system
- In Java compiler produce Bytecode for a Java Virtual Machine.
- When we compile a Java program, then bytecode is generated.
- You get this bytecode by compiling the .java files into .class files.
- .class files contain the bytecodes understood by the JVM.
- Bytecode is the source code that can be used to run on any platform.
- Bytecode is an intermediary language between Java source and the host system.

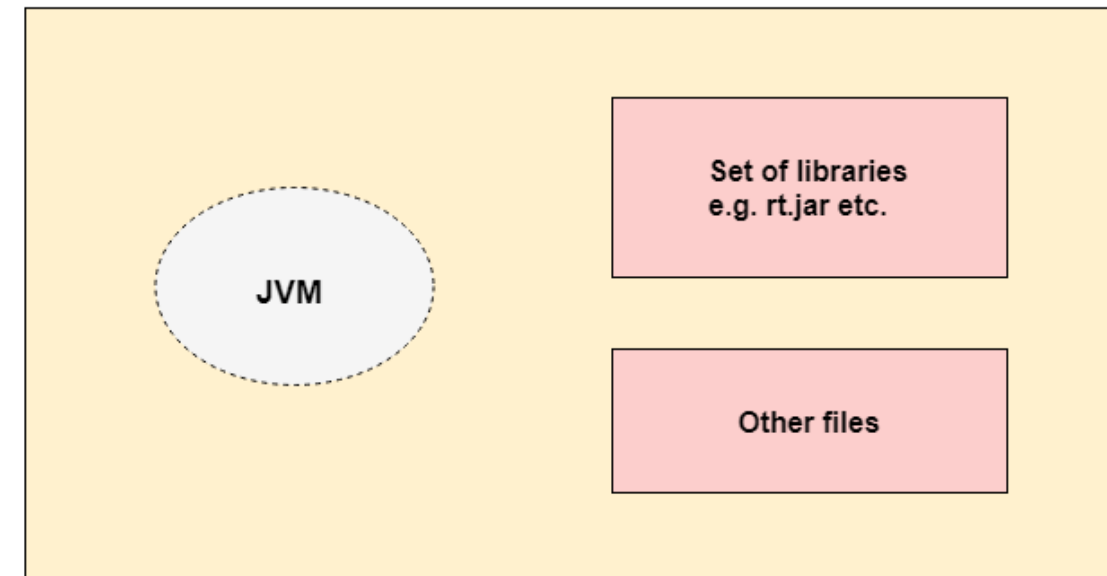
- JVM Provides a runtime environment in which Java bytecode can be executed.

The JVM performs the following main tasks:

- Loads code
- Verifies code
- Executes code
- Provides runtime environment

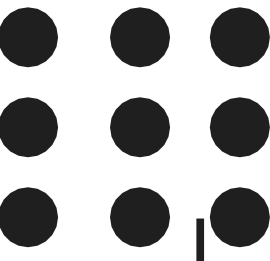
JRE

- The Java Runtime Environment (JRE) is a software package which bundles the libraries (jars) and the Java Virtual Machine, and other components to run applications written in the Java.
- JVM is just a part of JRE distributions.
- The Java Runtime Environment is a set of software tools which are used for developing Java applications.





JDK



- **JDK** is an acronym for **Java Development Kit**. The Java Development Kit (JDK) is a software development environment which is used to develop Java applications and applets.
- The JDK contains a private Java Virtual Machine (JVM) and a few other resources such as an interpreter/loader (java), a compiler (javac), an archiver (jar), a documentation generator (Javadoc), etc. to complete the development of a Java Application.
- JDK SE 16 is the latest version

IDE (Integrated Development Environment) Tools to run java

- Eclipse
- NetBeans
- IntelliJ IDEA
- Kite



Data Types



There are two data types available in Java

Primitive Data Types

The primitive data types include boolean, char, byte, short, int, long, float and double.

Reference/Object Data Types

The non-primitive data types include Classes, Interfaces, and Arrays.

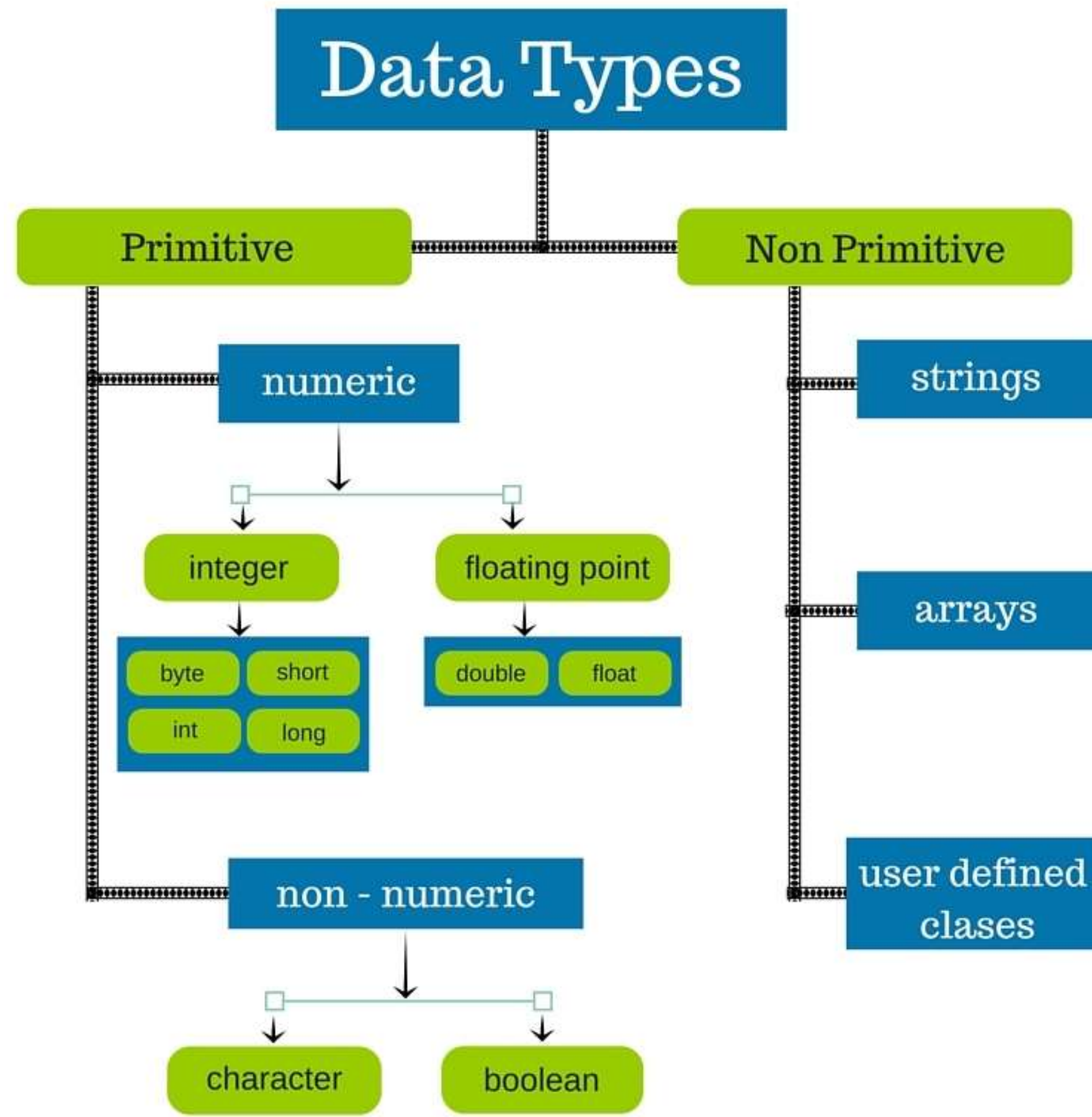
Specify the type of data and the length of the data item in bytes

int, short, long

float, double

boolean

char



Data Types

Reserved Word	Data Type	Size	Range of Values
byte	Byte Length Integer	1 bytes	-2^8 to $2^7 - 1$
short	Short Integer	2 bytes	-2^{16} to $2^{16} - 1$
int	Integer	4 bytes	-2^{32} to $2^{31} - 1$
long	Long Integer	8 bytes	-2^{64} to $2^{63} - 1$
float	Single Precision	4 bytes	-2^{32} to $2^{31} - 1$
double	Real number with double	8 bytes	-2^{64} to $2^{62} - 1$
char	Character (16 bit unicode)	2 bytes	0 to 216 - 1
boolean	Has value true or false	A boolean value	true or false



Keywords



- Java keywords are also known as reserved words.
- Keywords are particular words which acts as a key to a code
- These are predefined words by Java so it cannot be used as a variable or object name.

Keywords

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



Operators



Java provides a rich set of operators to manipulate variables.

We can divide all the Java operators into the following groups

- Arithmetic Operators
- Relational Operators
- Bitwise Operators
- Logical Operators
- Assignment Operators
- Unary Operators
- Ternary Operators
- Misc Operators

Operators

Operators	Associativity	Type
++ --	Right to left	Unary postfix
++ -- + - ! (type)	Right to left	Unary prefix
/ * %	Left to right	Multiplicative
+ -	Left to right	Additive
< <= > >=	Left to right	Relational
== !=	Left to right	Equality
&	Left to right	Boolean Logical AND
^	Left to right	Boolean Logical Exclusive OR
	Left to right	Boolean Logical Inclusive OR
&&	Left to right	Conditional AND
	Left to right	Conditional OR
?:	Right to left	Conditional
= += -= *= /= %=	Right to left	Assignment



THANK YOU