

$$\int_{\Omega} \nabla u \cdot \nabla v \, dx = \int_{\partial\Omega} u \nabla v \cdot \nu \, d\sigma - \int_{\Omega} u \Delta v \, dx = \iint_D \left( \frac{\partial M}{\partial x} - \frac{\partial L}{\partial y} \right) dA$$

$$\int_b^c f(x)g'(x) \, dx = \left[ f(x)g(x) \right]_b^c - \int_b^c f'(x)g(x) \, dx$$

$$f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \dots$$

# JAVA REFRESHER COURSE

## Part I



Muhammad Sadiq Sarfaraz  
Institute of Scientific Computing  
Technical University Braunschweig

- Introduction
- Why learn Java
- Java virtual machine
- Data and variable types
- Operators
- Loops and Conditional statements
- Arrays and Strings

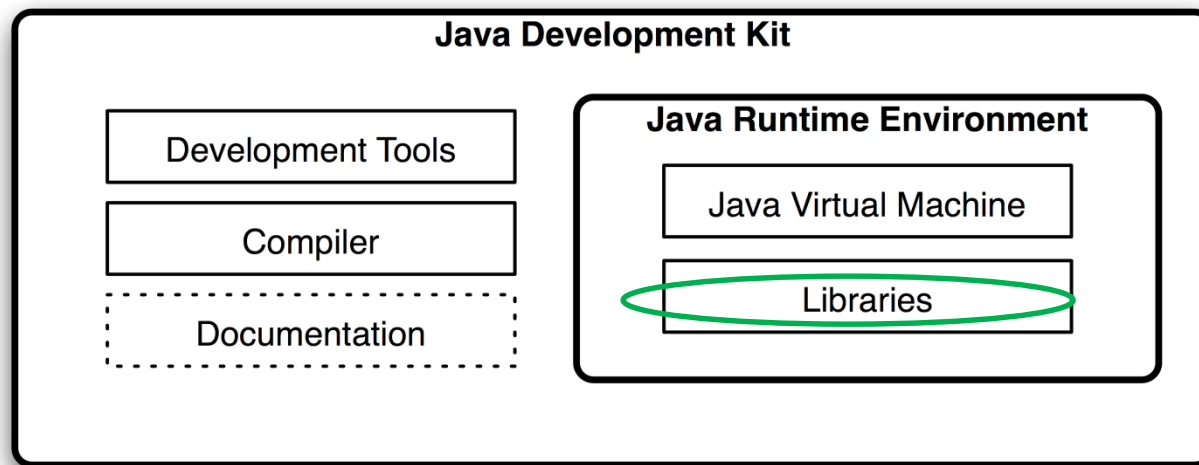
- Java is a high-level programming language originally developed by Sun Microsystems and released in 1995.
- Platform Independent :Java runs on a variety of platforms,
- High Level language
- Object Oriented Language

ORACLE®

- **Why Java?**
- Object oriented
- Interpreted
- Portable
- Simple yet feature-full
- Secure and robust
- Scalable
- High-performance multi-threaded
- Dynamic
- Distributed

- **JVM – JAVA VIRTUAL MACHINE**

- JVM is an abstract machine which provides run time environment for the execution of Java program. As JVM is abstract, it can be implemented specifically for underlying platform (OS and hardware architecture).



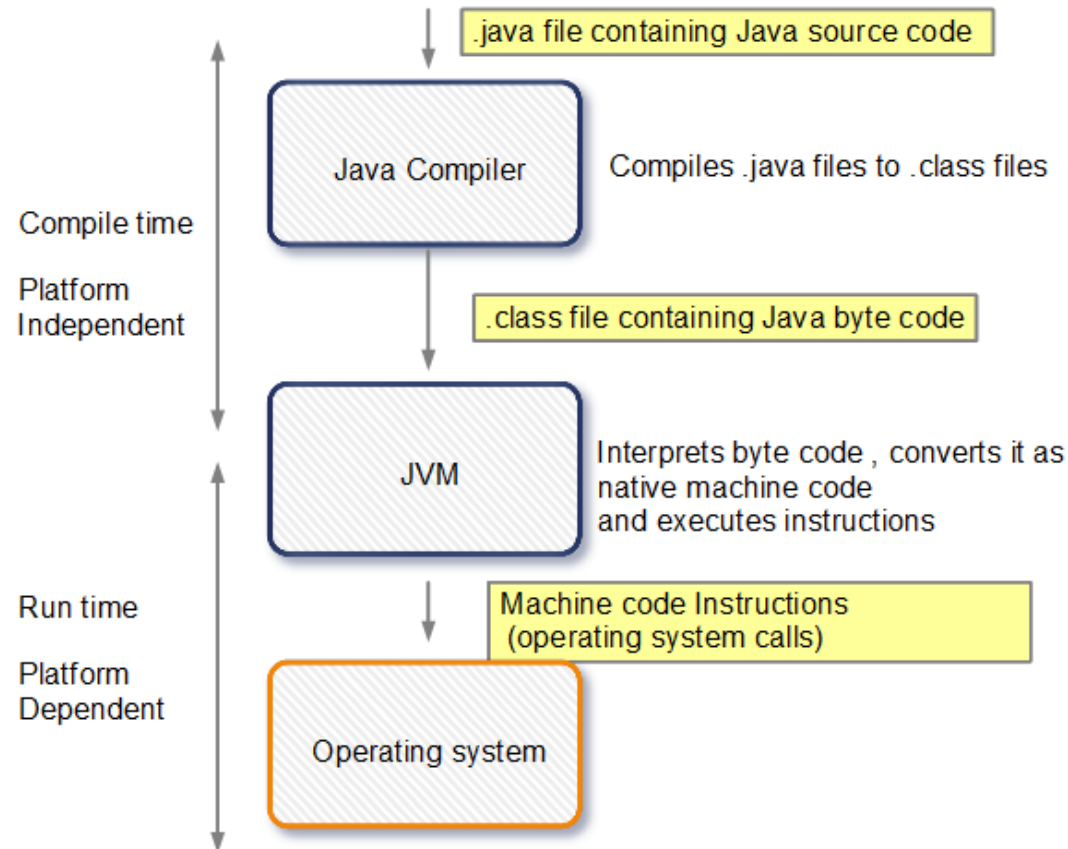
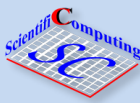
- The JDK comes with a command-line compiler: `javac`.
- It compiles source code into *Java bytecode*, which is low-level instruction set similar to binary machine code.
- The bytecode is executed by a *Java virtual machine (JVM)*, rather than a specific physical processor.

$$\int_{\Omega} \nabla u \cdot \nabla v \, dx = \int_{\partial \Omega} u \nabla v \cdot \nu \, d\sigma - \int_{\Omega} u \Delta v \, dx = \iint_D \left( \frac{\partial M}{\partial x} - \frac{\partial L}{\partial y} \right) dA$$

$$\int_b^c f(x) g'(x) \, dx = \left[ f(x) g(x) \right]_b^c - \int_b^c f'(x) g(x) \, dx$$

$$f(x) = f(a) + \frac{f'(a)}{1!} (x-a) + \frac{f''(a)}{2!} (x-a)^2 + \frac{f'''(a)}{3!} (x-a)^3 + \dots$$

# Java Code Execution



- There are two data types available in Java:

## 1. Primitive Data Types

Primitive data types are predefined by the language and named by a key word. There are 8 primitive data types in java

## 2. Reference/Object Data Types

Reference variables are created using defined constructors of the classes. They are used to access objects. These variables are declared to be of a specific type that cannot be changed.

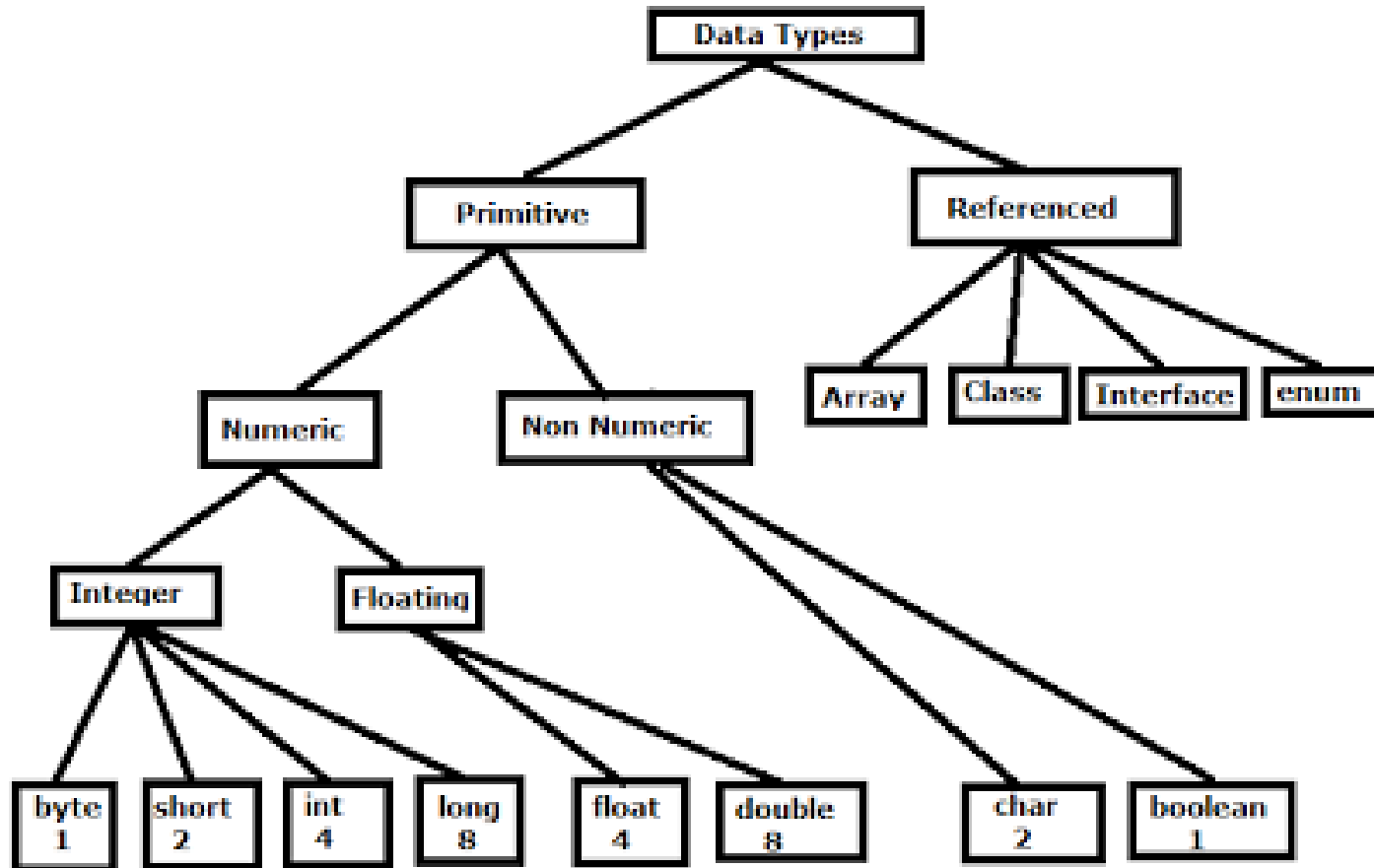


$$\int_{\Omega} \nabla u \cdot \nabla v \, dx = \int_{\partial\Omega} u \nabla v \cdot \nu \, d\sigma - \int_{\Omega} u \Delta v \, dx = \iint_D \left( \frac{\partial M}{\partial x} - \frac{\partial L}{\partial y} \right) dA$$

$$\int_b^c f(x)g'(x) \, dx = \left[ f(x)g(x) \right]_b^c - \int_b^c f'(x)g(x) \, dx$$

$$f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \dots$$

## Data Types in Java



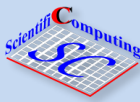
[www.instanceofjava.com](http://www.instanceofjava.com)

$$\int_{\Omega} \nabla u \cdot \nabla v \, dx = \int_{\partial\Omega} u \nabla v \cdot \nu \, d\sigma - \int_{\Omega} u \Delta v \, dx = \int_{\partial\Omega} \left( \frac{\partial M}{\partial x} - \frac{\partial L}{\partial y} \right) dA$$

$$\int_a^b f(x) g'(x) \, dx = [f(x) g(x)]_a^b - \int_a^b f'(x) g(x) \, dx$$

$$f(x) = f(a) + \frac{f'(a)}{1!} (x-a) + \frac{f''(a)}{2!} (x-a)^2 + \frac{f'''(a)}{3!} (x-a)^3 + \dots$$

# Size requirements for data types in Java



## 3.2. Java Primitive Types

Type	Size	Range	Default*
<b>boolean</b>	1 bit	<b>true</b> or <b>false</b>	<b>false</b>
<b>byte</b>	8 bits	[-128, 127]	<b>0</b>
<b>short</b>	16 bits	[-32,768, 32,767]	<b>0</b>
<b>char</b>	16 bits	['\u0000', '\uffff'] or [0, 65535]	'\u0000'
<b>int</b>	32 bits	[-2,147,483,648 to 2,147,483,647]	<b>0</b>
<b>long</b>	64 bits	$[-2^{63}, 2^{63}-1]$	<b>0</b>
<b>float</b>	32 bits	32-bit IEEE 754 floating-point	<b>0.0</b>
<b>double</b>	64 bits	64-bit IEEE 754 floating-point	<b>0.0</b>

- A variable provides us with named storage that our programs can manipulate. Each variable in Java has a specific type, which determines the size and layout of the variable's memory
- Examples
  - `int a, b, c; // Declares three ints, a, b, and c.`
  - `int a = 10, b = 10; // Example of initialization`
  - `byte B = 22; // initializes a byte type variable B.`
  - `double pi = 3.14159; // declares and assigns a value of PI.`
  - `char a = 'a'; // the char variable a is initialized with value 'a'`

- There are three kinds of variables in Java:

1. Local variables

Local variables are declared in methods, constructors, or blocks.

2. Instance variables

Instance variables are declared in a class, but outside a method, constructor or any block.

3. Class/static variables

Class variables also known as static variables are declared with the *static* keyword in a class, but outside a method, constructor or a block

- The following list shows the reserved words in Java. These reserved words may not be used as constant or variable or any other identifier names.

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

# 1. Arithmetic Operators

$+$ ,  $-$ ,  $*$ ,  $/$  ...

# 2. Relational Operators

$==$ ,  $!=$ ,  $>$ ,  $<$ , ...

# 3. Logical Operators

$\&\&$ ,  $||$ ,  $!$

# 4 Assignment Operators

$=$ ,  $+=$ ,  $-=$

# 5 Others : Bitwise, Conditional etc

- while loops

Repeats a statement or group of statements while a given condition is true. It tests the condition before executing the loop body.

- for loop

Execute a sequence of statements multiple times and abbreviates the code that manages the loop variable.

- do...while loop

Like a while statement, except that it tests the condition at the end of the loop body

- If statement

An if statement consists of a boolean expression followed by one or more statements.

- If... else statement

An if statement can be followed by an optional else statement, which executes when the boolean expression is false.

- switch statement

A switch statement allows a variable to be tested for equality against a list of values



- **array** stores a fixed-size sequential collection of elements of the same type. An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type.
- Declare array values:  
    `int[] values;`  
    `int values[];`
- Arrays can be single- or multi-dimensional.  
    `double values[][];`
- Access individual elements with *arrayName[index]*
- new operator to create an array of an indicated type and length  
    `int[] values = new int[10];`

- In Java, strings are instances of the String class. They are not array of chars
- Strings are *immutable*: Once created, a String cannot be modified. However manipulation can be performed using String methods yielding new objects
- Methods of String class: length, replace, substring, indexOf, equals, trim, split, toUpperCase, endsWith, etc.
- Example : `String s = "my string"; int len = s.length();`