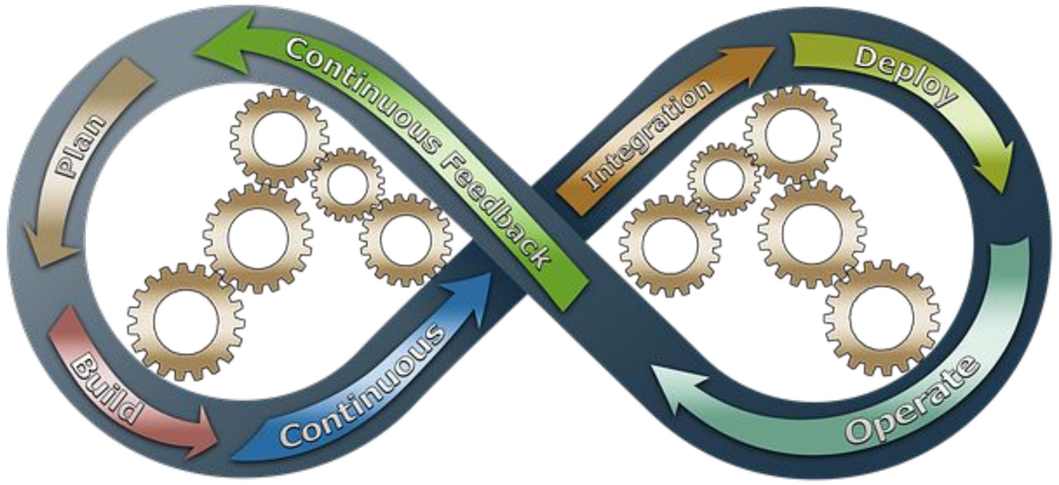


Session 3.3

Overview of Java

AN INITIATIVE BY

UNICAL ACADEMY

UNICAL ACADEMY

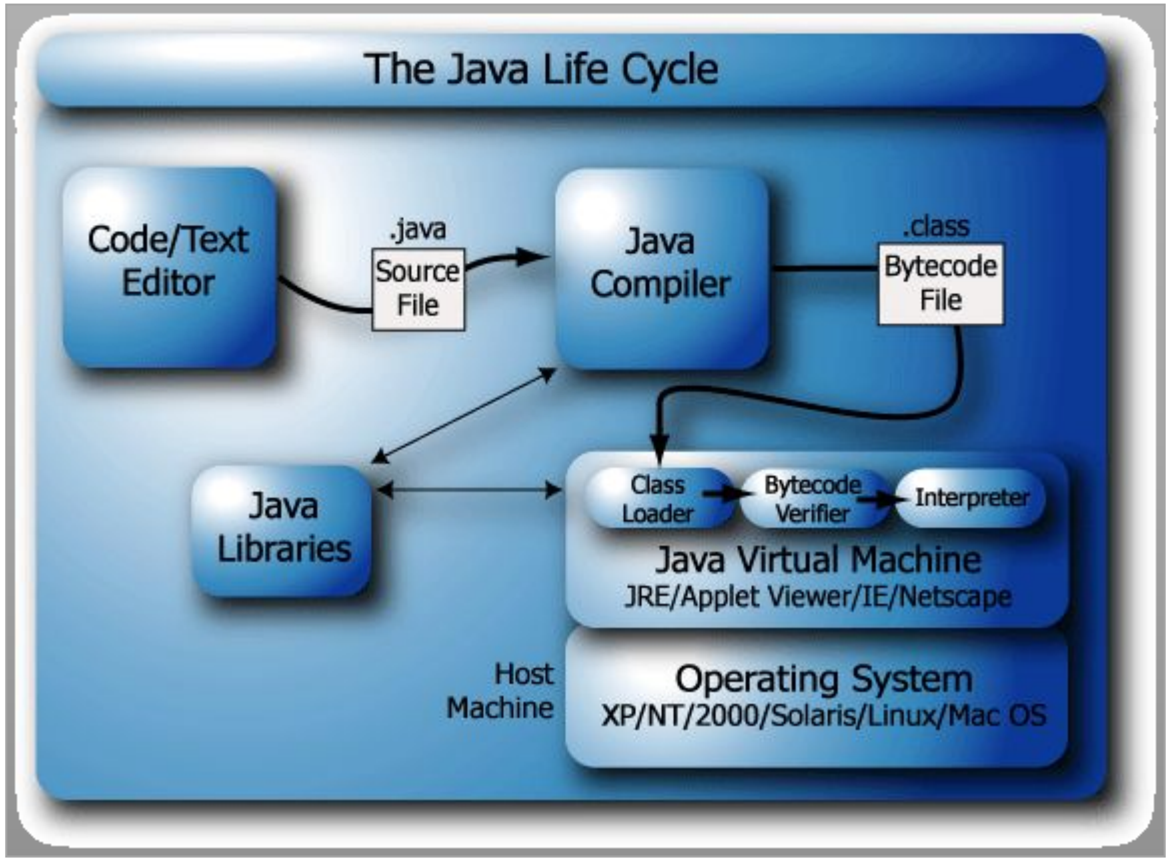
Introduction

Let’s go!!!

Overview of Java

<My Notes: Here, Key features of java, where java can be used as a programming language, History of various Java versions will be covered

Java has emerged as the Object-Oriented Programming Language and efficient for application programming. Some of the important concepts of Java include:



Historical Perspective of Java

History of various Java versions

1995 – JDK BETA
The first beta version of java. Developed by James Gosling at Sun Microsystems

1996 – JAVA 1
First public release. The stable version Java 1.02 is called Java 1

1997 – JAVA 1.1
Inner classes, Java Beans, JDBC, RMI

1998 – JAVA 1.2
Swing, JIT Compiler, Collections

2000 – JAVA 1.3
HotSpot JVM, JNDI, JPDA

2002 – JAVA 1.4
Assertions, RegEx Improvements, Image IO API, XML Parsers, XSLT Processors, Preferences API

2004 – JAVA 5
Generics API, Varargs, for-each loop, Autoboxing, Enum, Annotations, Static Imports

2006 – JAVA 6
JAXB 2, JDBC 4.0 support, Pluggable annotations

2011 – JAVA 7
String in Switch Statements, Try with Resource, Java NIO Package, Catching Multiple Exceptions in a single catch block

2014 – JAVA 8
For Each() method, default and static method in interfaces, Functional interfaces and Lambda expressions, Stream API, New Date Time API

2017 – JAVA 9
Jshell, Module System under Project jigsaw, Reactive Streams, HTTP 2 CLient

2018 – JAVA 10
Local-variable Type Interface

2018 – JAVA 11
Running Java program from single command, New String Class methods, var for lambda expressions

2019 – JAVA 12
The first beta version of java. Developed by James Gosling at Sun Microsystems

2019 – JAVA 13
Text Blocks, Switch Expressions, Socket API reimplementation, Unicode 12.1 support, DOM and SAX Factories with Namespace Support

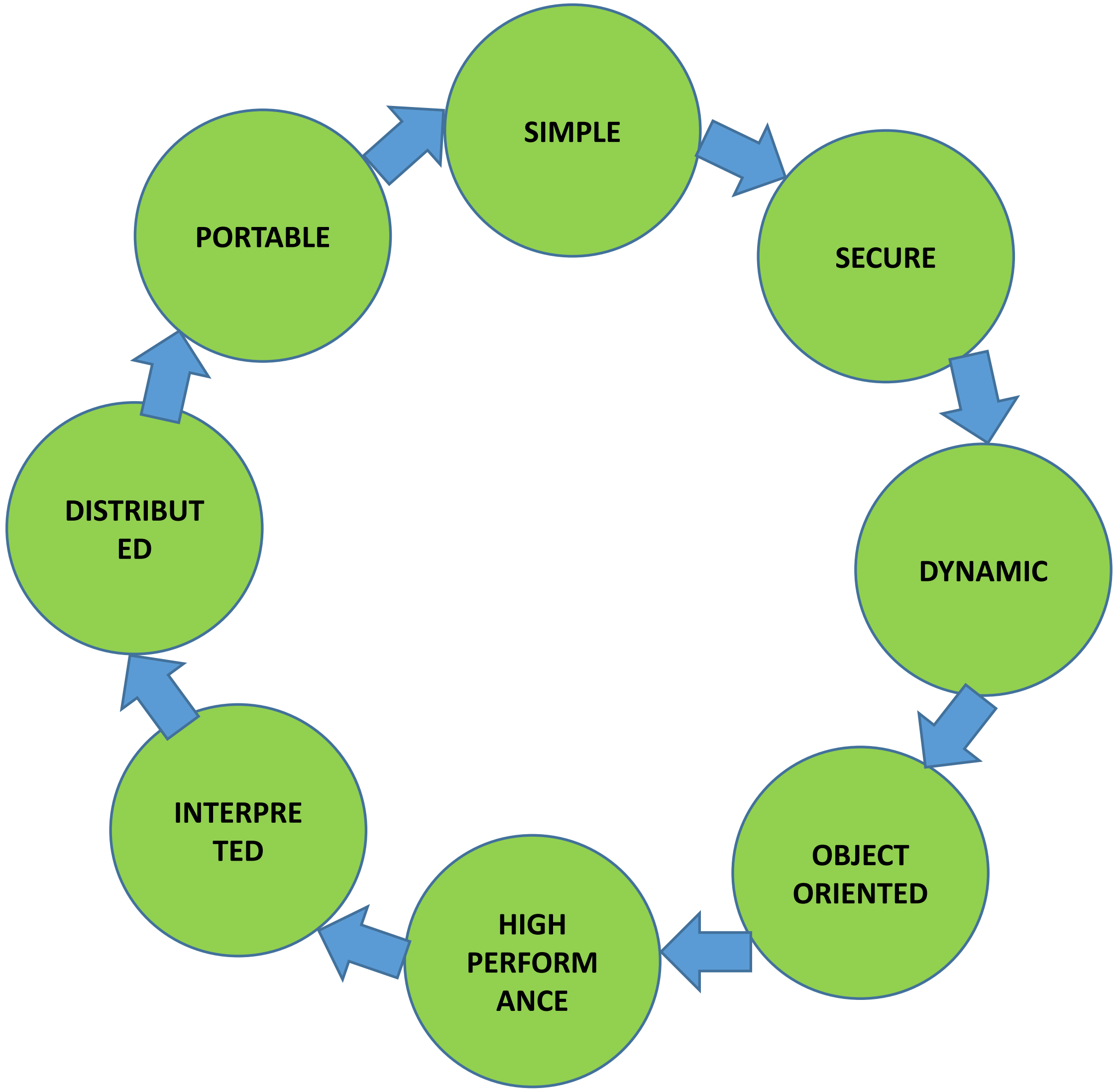
2020 – JAVA 14

2020 – JAVA 15

2021 – JAVA 16

UNICAL ACADEMY

Key Features of the Java Language –



Types of Java Applications:

- Standalone Application
- Web Application
- Enterprise Application
- Mobile Application

Java Platforms / Editions:

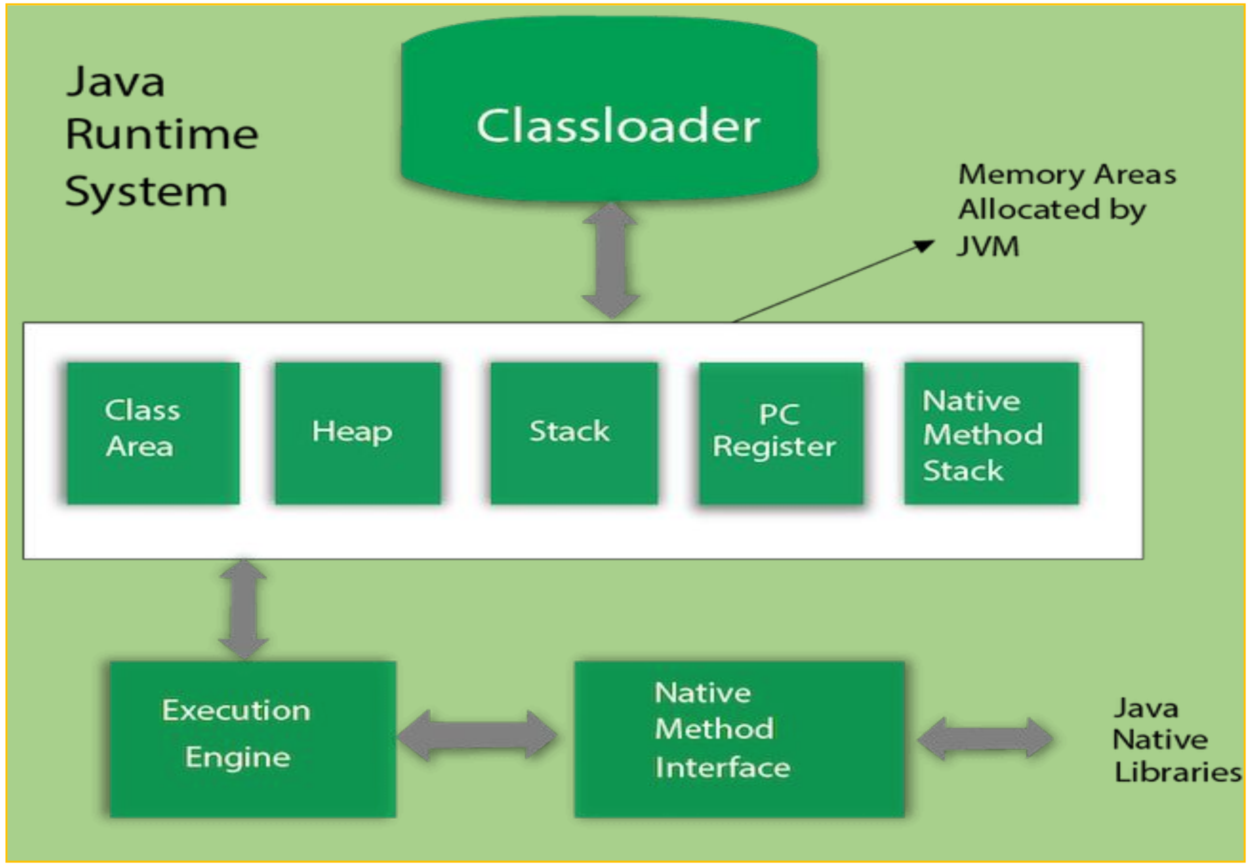
- Java SE (Java Standard Edition)
- Java EE (Java Enterprise Edition)
- Java ME (Java Micro Edition)
- JavaFX

Java Architecture

UNICAL ACADEMY

Java Architecture explains each and every step of how a program is compiled and executed. The Java architecture includes the three main components:

Java Virtual Machine (JVM)



JVM Architecture

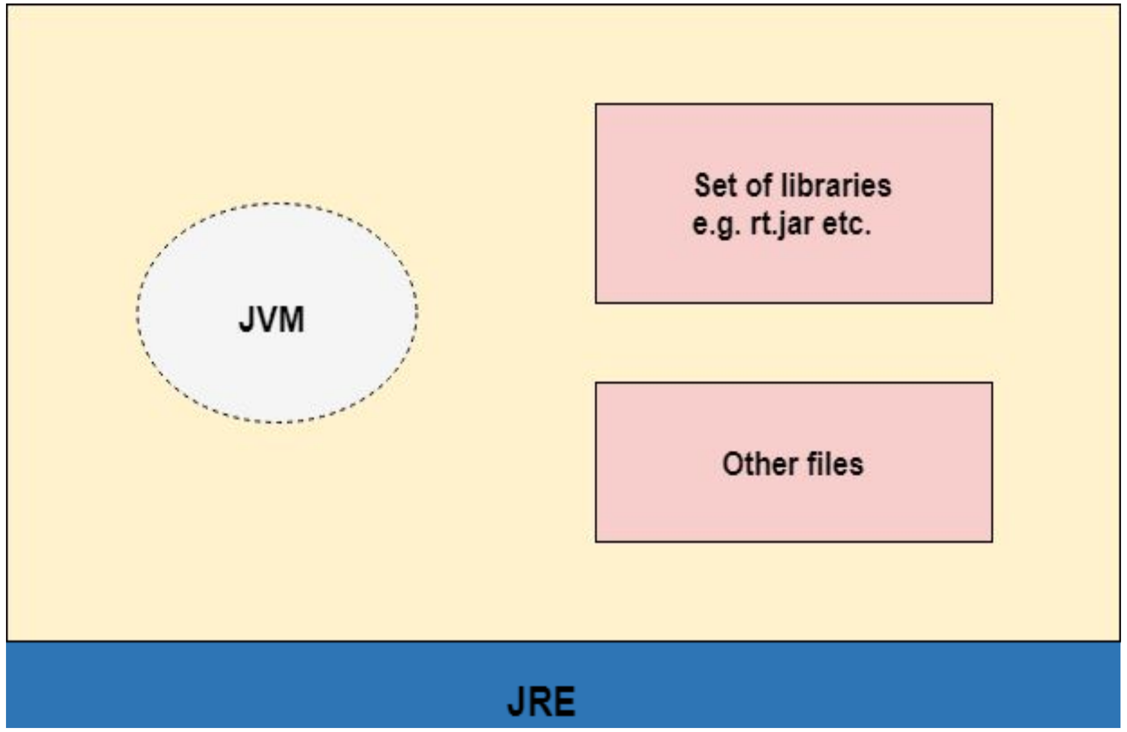
JVM Main Tasks:

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

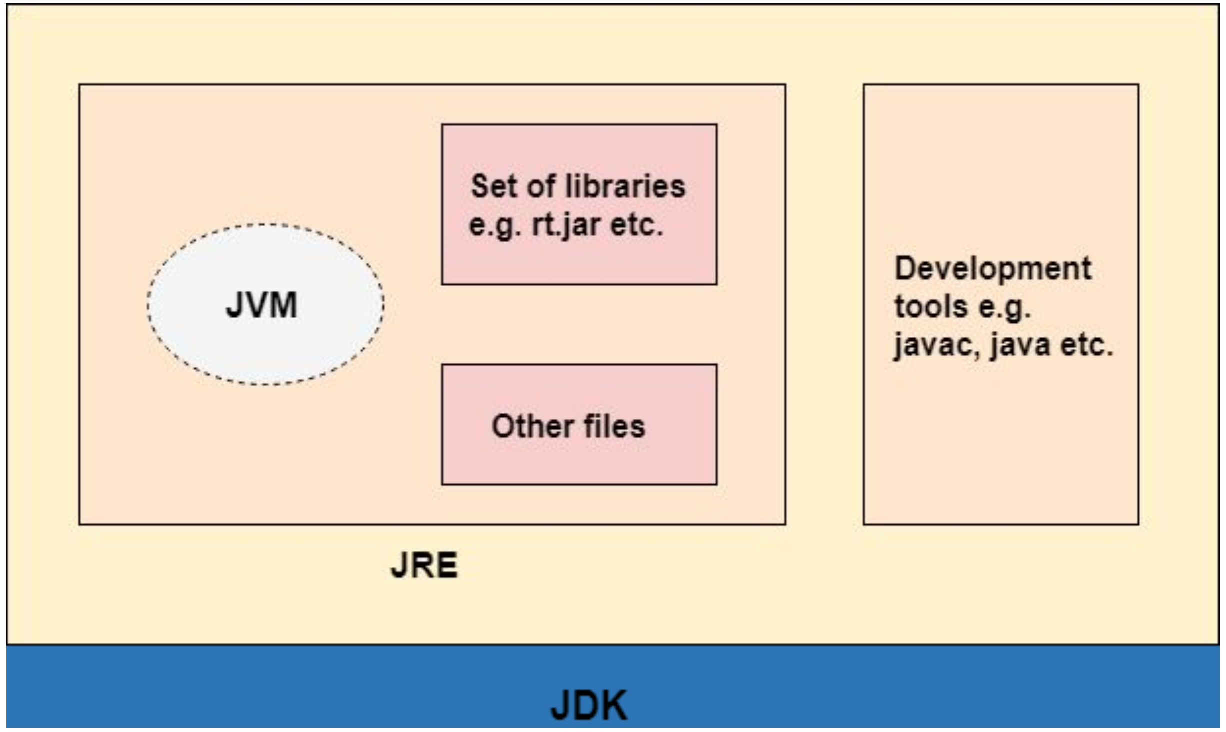
Java Architecture

UNICAL ACADEMY

Java Runtime Environment (JRE)



Java Development Kit (JDK)


JVM
Set of libraries e.g. rt.jar etc.
Other files
Development tools e.g. javac, java etc.
JRE
JDK

UNICAL ACADEMY

First Java Program & other simple Programs

<My Notes: Here, we will learn how to write the Hello World program and Other simple programs of java>

Let's explore how Java "Hello World" program works –

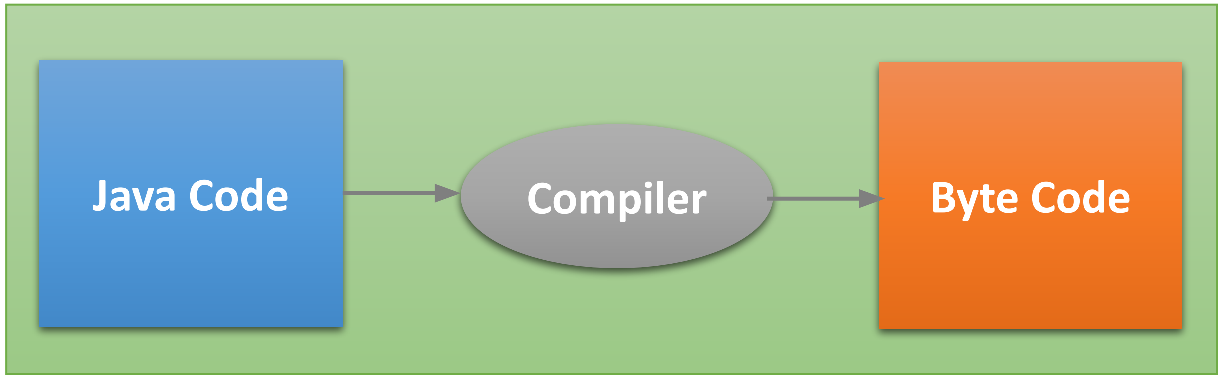
A "Hello World" is a simple program that outputs Hello World on the screen.

Pre-requisites for Java Program

- For executing any java program, you need to Install the JDK
- download the JDK and install it.
- Set path of the jdk/bin directory.
- http://www.javatpoint.com/how-to-set-path-in-java.
- Create the java program
- Compile and run the java program

UNICAL ACADEMY

Compilation Flow:



Parameters used in First Java Program

- class
- public
- static
- void
- main
- String[] args
- System.out.println()

<My Notes: Here we can learn parameters used to write java program and creating the Hello World program>

UNICAL ACADEMY

Creating First Java Program

```
// Your First Program

class HelloWorld
{
public static void main(String args[])
  {
  System.out.println("Hello World");
  }
}
```

Output: **Hello World**

How Java "Hello World" Program Works?

In Java, any line starting with// is a comment. Comments are intended for users reading the code to better understand the intent and functionality of the program.

In Java, every application begins with a class definition. In the program, HelloWorld is the name of the class.

It prints the text Hello World to standard output (your screen). The text inside the quotation marks is called String in Java.

This is the main method. Every application in Java must contain the main method. The Java compiler starts executing the code from the main method.

Printing an Integer entered by an user

<My Notes: >

```
import java.util.Scanner;
public class HelloWorld
{
public static void main(String[] args)
 {
 // Creates a reader instance which takes
// input from standard input - keyboard
Scanner reader = new Scanner(System.in);
System.out.print("Enter a number: ");
 // nextInt() reads the next integer from the keyboard
 int number = reader.nextInt();
// println() prints the following line to the output screen
 System.out.println("You entered: " + number);
  }
 }
```

In this program, an object of Scanner class, reader is created to take inputs from standard input, which is keyboard.

Then, Enter a number prompt is printed to give the user a visual cue as to what they should do next.

reader.nextInt() then reads all entered integers from the keyboard unless it encounters a new line character \n (Enter). The entered integers are then saved to the integer variable number.

If you enter any character which is not an integer, the compiler will throw an InputMismatchException.

Finally, number is printed onto the standard output (System.out) - computer screen using the function println().

Output:
```
Enter a number: 10
You entered: 10
```

Keywords and reserve words

- Keywords are predefined which have a unique meaning and functionality in Java Programming Language.
- These keywords are also known as reserved keywords which mean they cannot be used as a variable name, class, method or any other identifier.

Keywords in Java	**Description**
abstract	A class that is declared with abstract keyword, is known as abstract class in java. It can have abstract and non-abstract methods (method with body).
continue	It is is allowed only inside a loop body. When continue executes, the current iteration of the loop body terminates, and execution continues with the next iteration of the loop.
for	For is used for looping. It involves initialization, a boolean expression, and incrementation / decrementation. It supports repeated execution of a statement or block of statements that is controlled by an iterable expression.
new	used to create an instance of a class, or an object.
switch	Used as a statement which executes when it matches to a specific case.
assert	Assert keyword is added in 1.4 version. It describes a predicate (true-false statement), to let developers think that it's always true. If an assertion is false at run-time, it causes execution to abort.
default	Used in a switch statement to execute a block of code in the loop.
goto	goto has no function and it is no more supported in Java programming.
package	package is a mechanism of grouping similar type of classes, interfaces, and sub-classes collectively based on functionality.

Keywords and reserve words

UNICAL ACADEMY

synchronized	Synchronized blocks in Java are marked with the Synchronized keyword. This block in Java is synchronized on some object. All blocks that are synchronized on the same object can only have one thread executing inside them at a time.
boolean	boolean can hold true or false value only.
do	It is used in control statements. The Java do-while loop is used to iterate a set of statements until the given condition is satisfied.
if	If statement is used to test an expression and execute certain statements accordingly. It is also used to create an if-else statement in java.
private	Private is an access modifier in java, where the methods or data members that are declared as private are only accessible within the class in which they are declared.
this	this keyword in Java represents the current instance of a class. It is mainly used to access other members of the same class.
break	The break statement is allowed only inside a loop body. When break executes, the loop terminates.
double	It declares a variable that can hold 64-bit double floating-point numbers.
implements	used by a class to implements an interface.
protected	The methods or data members that are declared as private are only accessible within the class in which they are declared.
throw	used to create and throw an exception.
byte	It is used to declare a field which can hold 8-bit data values.
else	It is used to implement a condition alternate to if condition.

Keywords and reserve words

UNICAL ACADEMY

import	Used in the beginning which refers to other classes
throws	Used in method declarations which specifies exceptions that can’t be handled within the method.
case	used in the switch statements which can be labeled with one or more case
enum	Enum is added in 5.0 version.
instanceof	It evaluates to true if and only if the runtime type of the object is compatible with the class or interface.
return	Used to finish the execution of a method. It returns the value required by the method.
transient	It declares an instance field which is not a part of the default serialized form of an object.
catch	Statements in the catch block specify the exceptions generated by try block.
extends	Merely indicates that a class has extended its immediate class.
int	A data type that holds 32 bit signed integer.
short	A data type that holds a 16-bit integer.
try	It tests a block of code for exceptions.
public	Classes, methods or data members which are declared as public are accessible anywhere throughout the program. There is no restriction on the scope of public data members.
final	Once a certain entity is defined, it cannot be changed nor derived from later.
interface	Interface in Java refers to the abstract data types. They allow Java collections to be manipulated independently from the details of their representation.

Keywords and reserve words

static	static keyword is mainly used for memory management. It can be used with variables, methods, blocks and nested classes.
void	It returns a null value for a method.
char	It is a data type that can hold a 16-bit unsigned integer.
class	It creates a new class in Java which is a blueprint from which an object is created.
finally	It specifies that a block of code under exception handling always gets executed.
long	Data type holding a 64 bit integer.
strictfp	strictfp keyword is added in 1.2 version.
volatile	Specifies or indicates that a variable might change asynchronously.
const	This const java keyword is no more used.
float	A data type holding a 32-bit floating point number.
native	It specifies that a method declaration has tobe done from platform-specific(native) code.
super	super keyword refers to the members such as variable,method and constructor of of immediate super class.
while	It is used to create while loop. The Java while loop is used to iterate a part of the program again and again. If the number of iteration is not fixed, then you can use while loop.

Source code, Compile and Execute a Java program

UNICAL ACADEMY

Source code

Source code is the set of instructions and statements written by a programmer using a computer programming language. It contains declarations, functions, loops and other statements, which act as instructions for the program on how to function.

Compile

The compilation of the program is necessary to translate the program into a sequence of commands that can be directly executed by the computer. This process produces an intermediate object file. The standard Java compiler, which is part of the Java Standard Development Kit (Java SDK), is javac.

Link

Compiled languages come with library routines which can be added to the program. Theses routines are written by the manufacturer of the compiler to perform a variety of tasks, so even the most basic program will require a library function. After linking the file extension is .exe which are executable files.

Execute

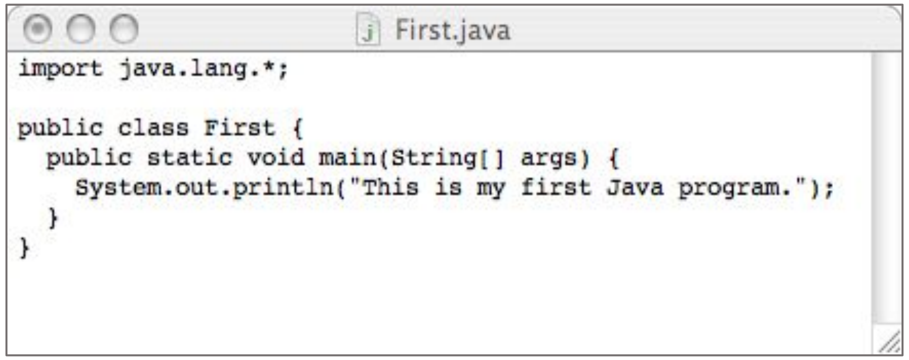
Thus the text editor produces .java source files, which go to the compiler, which produces .class object files, which go to the linker, which produces .exe executable file. You can then run .exe files as you can other applications.

UNICAL ACADEMY

Source code, Compile and Execute a Java program

Preparation of the program (Source code)

To prepare the program text we have to write a file containing the program. For a Java program, the name of the file has to be
ClassName.java
where *ClassName* is the name of the class defined in the program. E.g., First.java


First.java

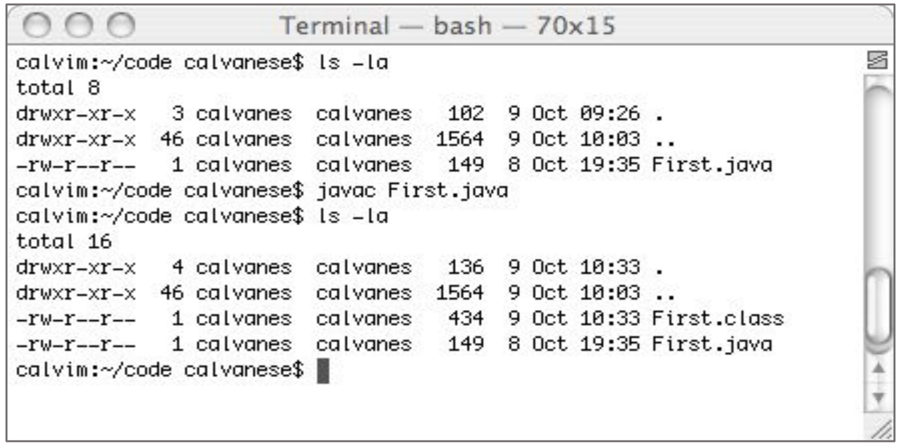
```
import java.lang.*;

public class First {
  public static void main(String[] args) {
    System.out.println("This is my first Java program.");
  }
}
```

UNICAL ACADEMY

Source code, Compile and Execute a Java program

Compilation of the program (Object code)

The compilation produces as a result a file called ClassName.class, which contains the command that can be directly executed by the computer.
For example:
javac First.java creates the file First.class.



```
calvim:~/code calvanese$ ls -la
total 8
drwxr-xr-x    3 calvanes  calvanes   102  9 Oct 09:26 .
drwxr-xr-x   46 calvanes  calvanes  1564  9 Oct 10:03 ..
-rw-r--r--    1 calvanes  calvanes   149  8 Oct 19:35 First.java
calvim:~/code calvanese$ javac First.java
calvim:~/code calvanese$ ls -la
total 16
drwxr-xr-x    4 calvanes  calvanes   136  9 Oct 10:33 .
drwxr-xr-x   46 calvanes  calvanes  1564  9 Oct 10:03 ..
-rw-r--r--    1 calvanes  calvanes   434  9 Oct 10:33 First.class
-rw-r--r--    1 calvanes  calvanes   149  8 Oct 19:35 First.java
calvim:~/code calvanese$ 
```

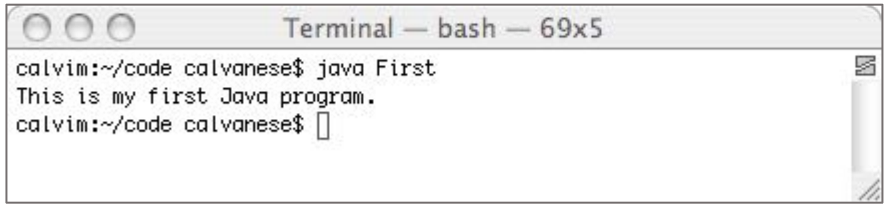
UNICAL ACADEMY

Source code, Compile and Execute a Java program

Execution of the compiled program (Executable Code)

A program can be executed only after it has been compiled, i.e., when we have the file *ClassName*.class.
In Java the execution of a program is done through the command
java *ClassName*

(without .class). For example, the command
java First
causes the execution of the program First (or, more precisely, of the main method of the class First), and hence prints on the screen:
This is my first Java program.



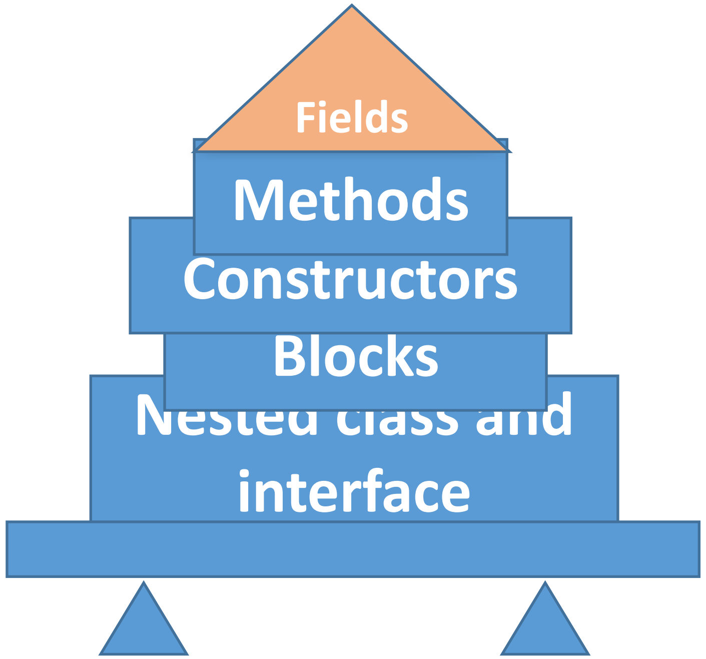
```
calvim:~/code calvanese$ java First
This is my first Java program.
calvim:~/code calvanese$ 
```

Class

UNICAL ACADEMY

Class

A class in Java contain:



Syntax to declare a class:

```
class <class_name>{
   field;
   method;
}
```

Object

Object

Characteristics of Object

State
Represents the data of an object

Behavior
Represents the behavior of an object such as Deposit, withdraw, etc..

Identity
It is used internally by the JVM to identify each object uniquely

Ways to create an Object

01 By new keyword

02 By newInsatance() method

03 By clone() method

04 By deserialization

05 By factory method etc.

Identifiers

Rules for Naming an Identifier

- Identifiers cannot be a keyword.
- Identifiers are case-sensitive
- It can have a sequence of letters and digits. However, it must begin with a letter, $ or _
- The first letter of an identifier cannot be a digit
- It's a convention to start an identifier with a letter rather and $ or _
- Whitespaces are not allowed
- Similarly, you cannot use symbols such as @, #, and so on.

Example:

```
public class Test
{
public static void main(String[] args)
{
int a = 20;
}
}
```

In the above java code, we have 5 identifiers namely :

- **Test** : class name
- **main** : method name
- **String** : predefined class name
- **args** : variable name
- **a** : variable name

Examples of Valid Identifiers

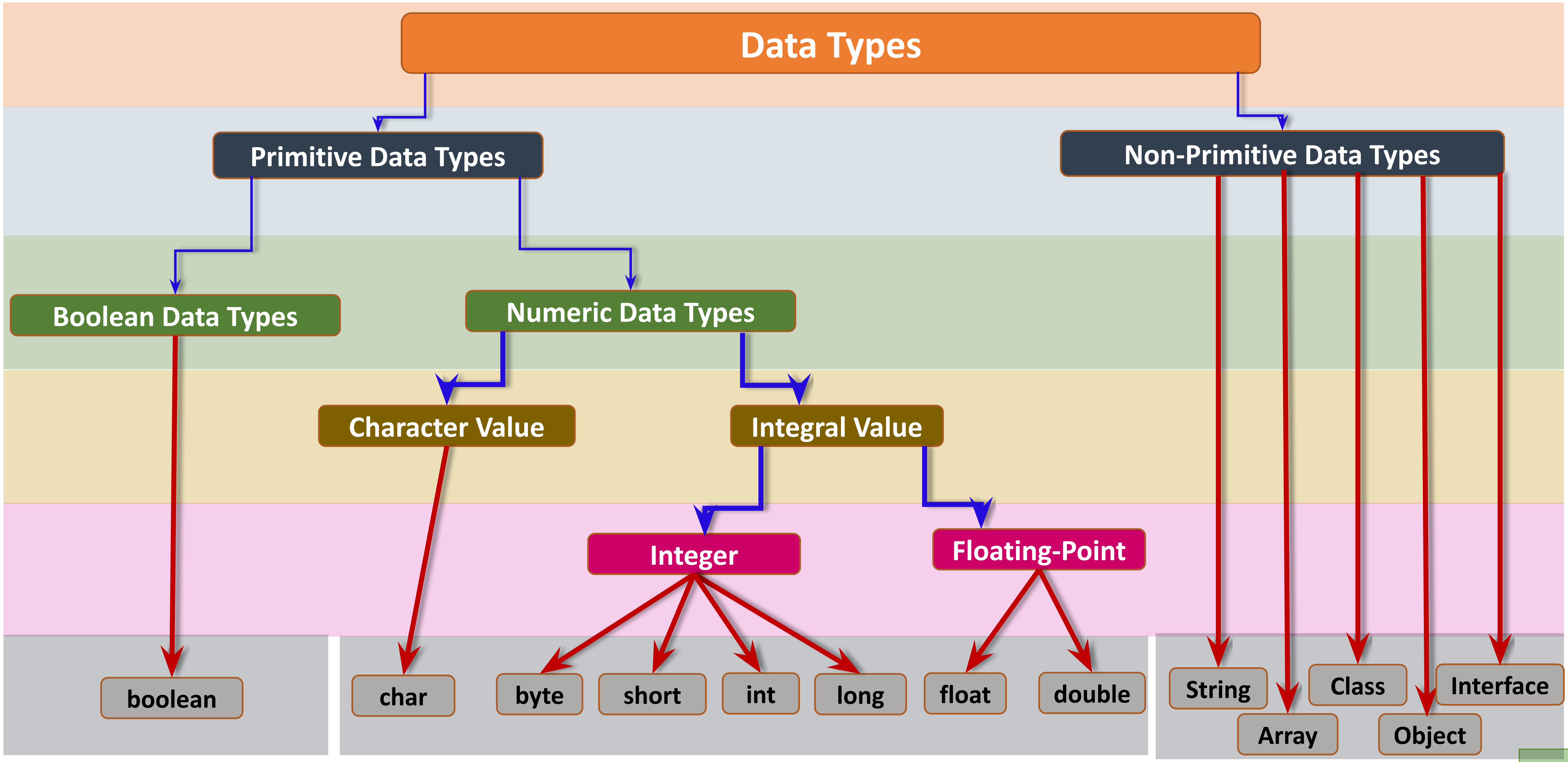
- MyVariable
- MYVARIABLE
- myvariable
- x
- i
- x1
- i1
- _myvariable
- $myvariable
- sum_of_array
- geeks123

Examples of Invalid Identifiers :

- My Variable // contains a space
- 123geeks // Begins with a digit
- a+c // plus sign is not an alphanumeric character
- variable-2 // hyphen is not an alphanumeric character
- sum_&_difference // ampersand is not an alphanumeric character

Data Types

UNICAL ACADEMY



UNICAL ACADEMY

Primitive Data Types

Type	Description	Default	Size	Range	Example Literals	Syntax
boolean	true/false	false	1 bit	true, false	true, false	boolean booleanVar;
byte	twos complement integer	0	8 bits	- 128 to 127	(none)	byte byteVar;
char	unicode character	\u0000	16 bits	character representation of ASCII values 0 to 255	'a', '\u0041', '\101', '\n', 'β'	char charVar;
short	twos complement integer	0	16 bits	- 32,768 to 32,767	(none)	short shortVar;
int	twos complement integer	0	32 bits	-2,147,483,648 to 2,147,483,647	-2, -1, 0, 1, 2	int intVar;
long	twos complement integer	0	64 bits	- 9,223,372,036,854,775,808 to 9,223,372,036,854,775,808	-2L, -1L, 0L, 1L, 2L	long longVar;
float	IEEE 754 floating point	0.0	32 bits	Upto 7 decimal digits	1.23e100f, -1.23e-100f, .3f, 3.14F	float floatVar;
double	IEEE 754 floating point	0.0	64 bits	Upto 16 decimal digits	1.23456e300d, -1.23456e-300d, 1e1d	double doubleVar;

UNICAL ACADEMY

Non-Primitive or Reference Data Types

The **Non-Primitive or Reference Data Types** will contain a memory address of variable value because the reference types won't store the variable value directly in memory.

String

Syntax:

```
<String_Type> <string_variable> = "<sequence_of_string>";
```

Example:

```
// Declare String without using new operator
 String s = "GeeksforGeeks";

// Declare String using new operator
 String s1 = new String("GeeksforGeeks");
```

UNICAL ACADEMY

Non-Primitive or Reference Data Types

01 Non-primitive types are created by the programmer and is not defined by Java (except for String).

02 Non-primitive types can be used to call methods to perform certain operations.

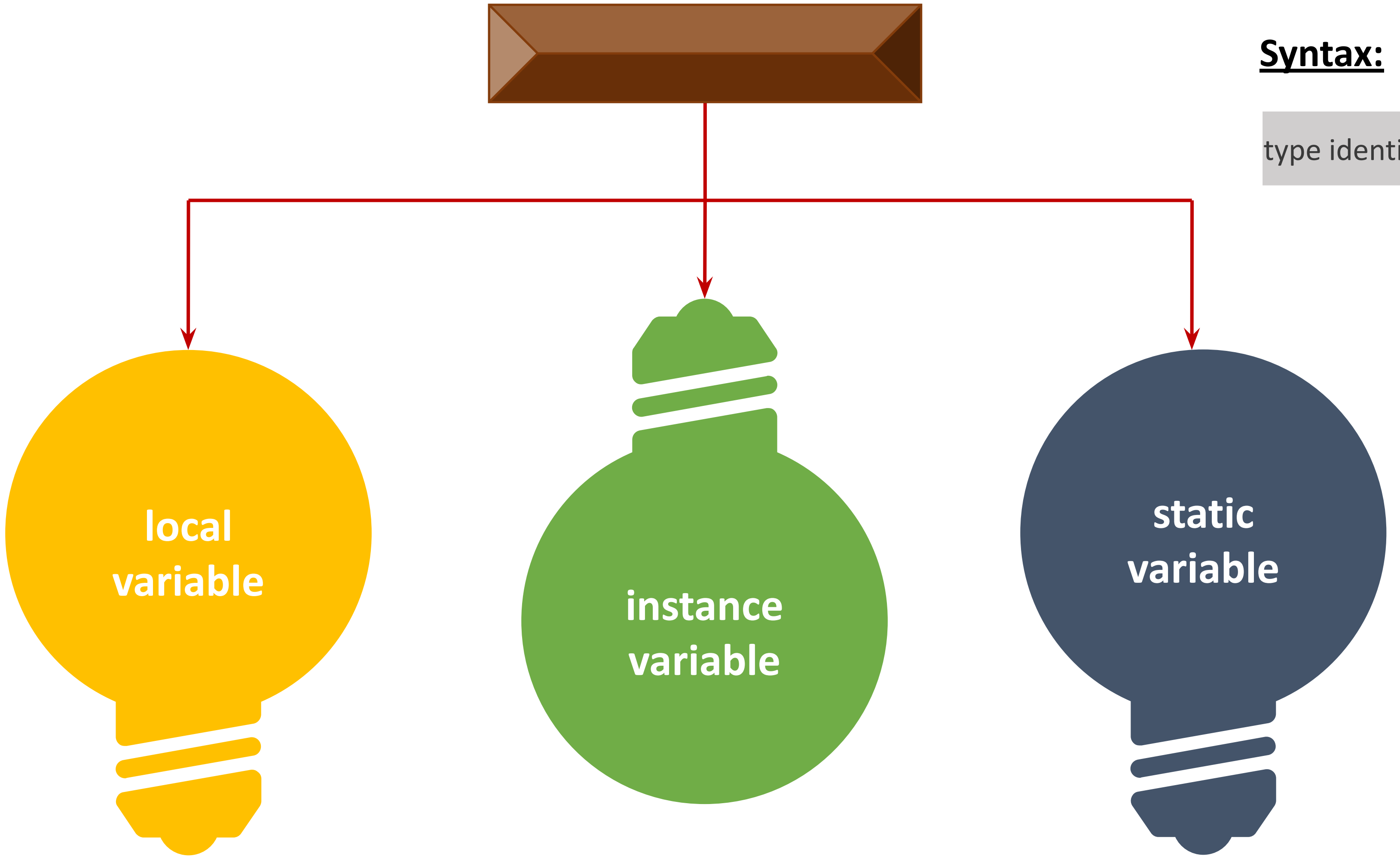
03 A non-primitive type has always null value.

04 A non-primitive type starts with an uppercase letter.

05 The size of a non-primitive types have all the same size.

Variable Types

UNICAL ACADEMY

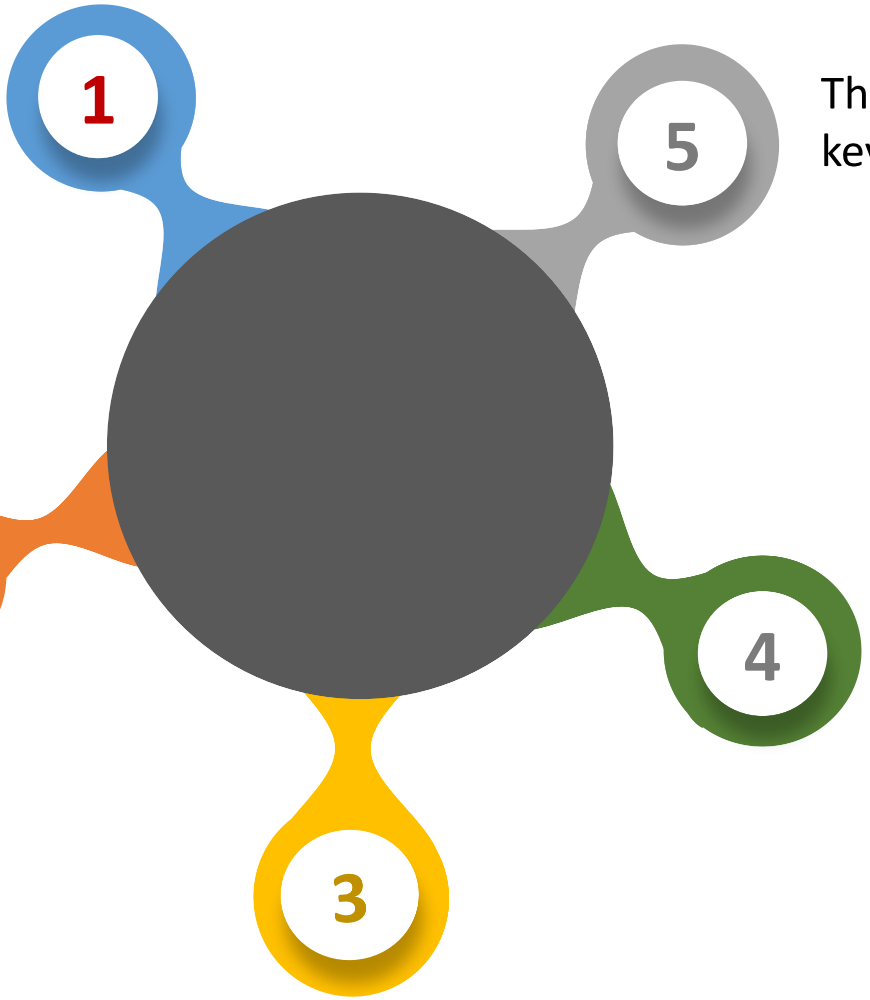


Syntax:

```
type identifier [ = value][, identifier [= value] ...]
```

UNICAL ACADEMY

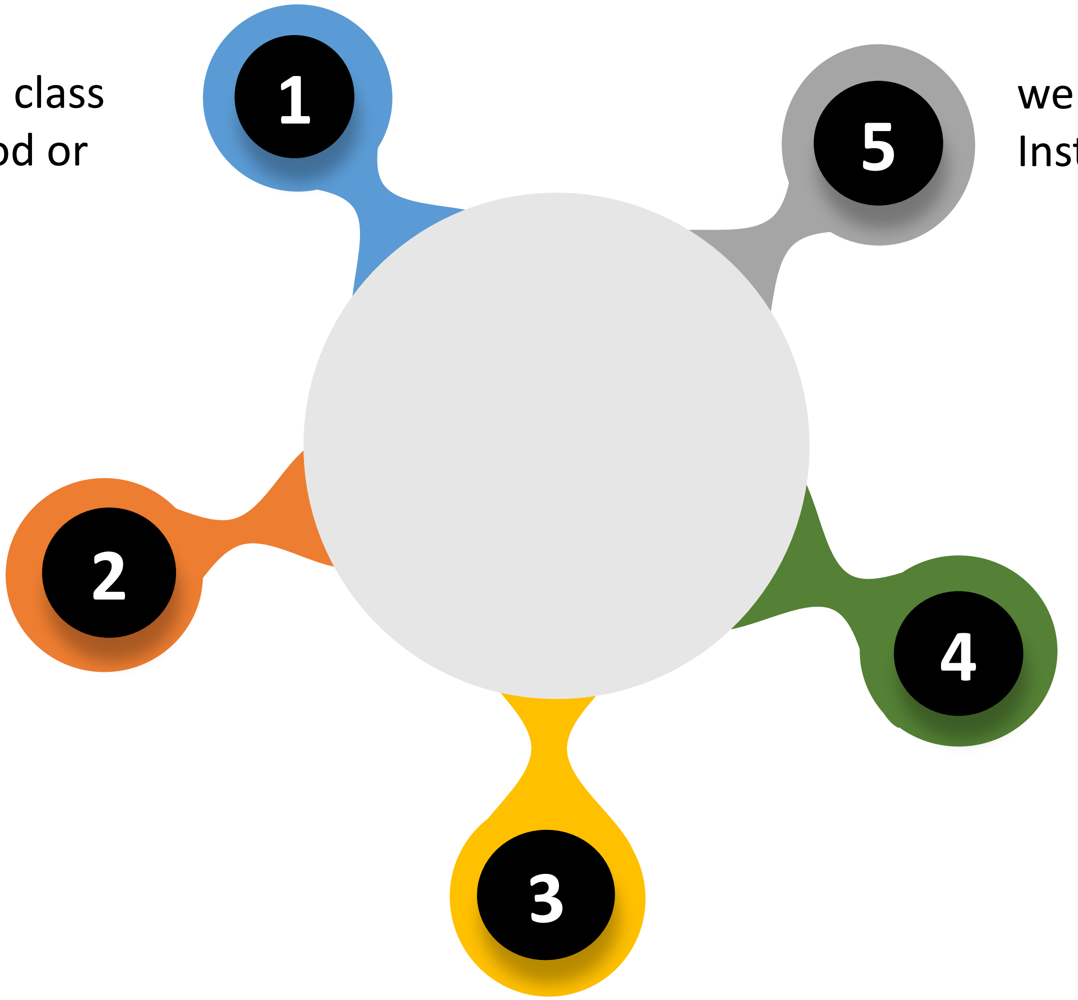
Local Variable Properties:



1. Local variables are declared inside a method, constructor, or block
2. No access modifiers for local variables
3. These can be used only within the same block, method, or constructor where it is initialized
4. The local variables are not initialized to any default value. We need to initialize declared local variable.
5. These can't be defined by a static keyword

UNICAL ACADEMY

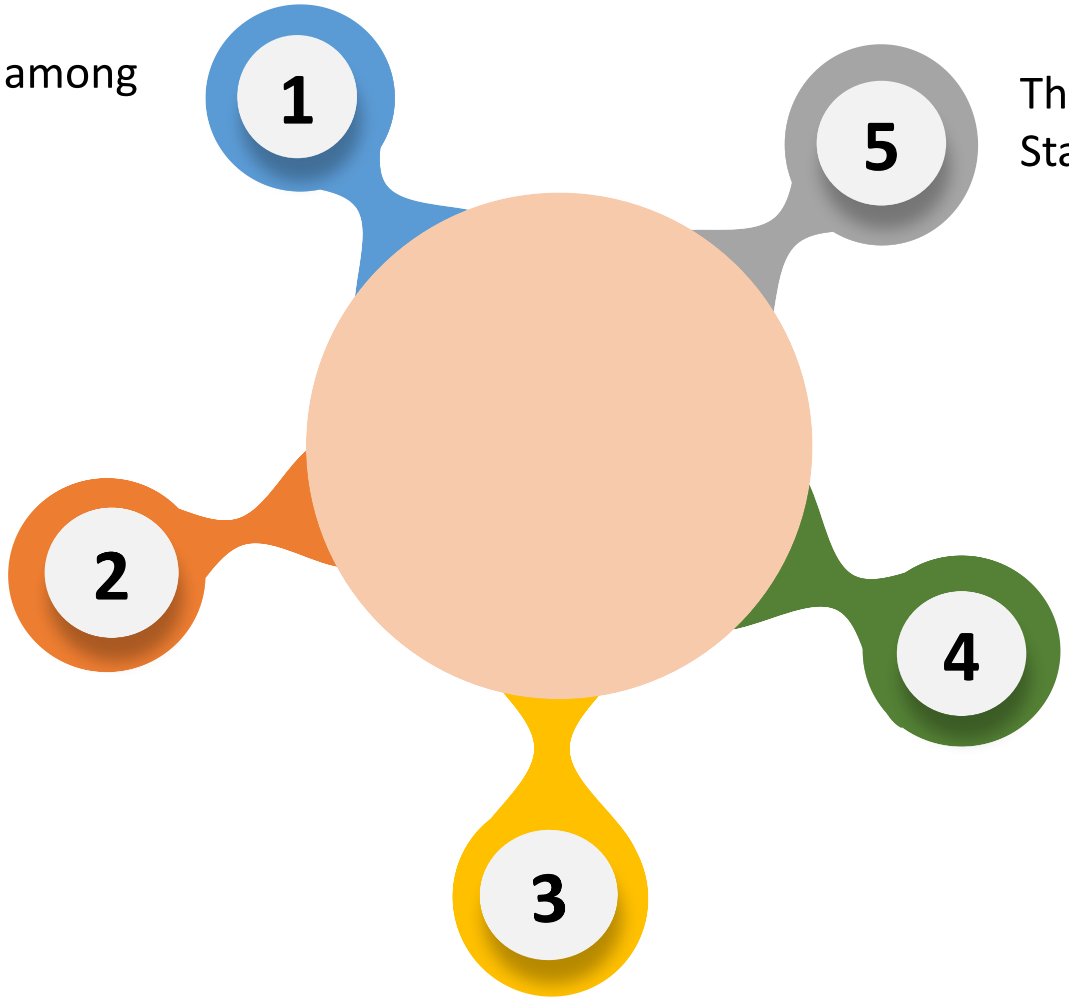
Instance Variable Properties:



1. They are declared within a class but outside a block, method or constructor.
2. It cannot be defined by a static keyword.
3. These variables have a default value
4. The integer type has a default value '0' and the boolean type has the default value 'false'.
5. we have access modifiers for Instance variables.

UNICAL ACADEMY

Static / Class Variable Properties:



1. static variables are shared among all the instances of a class.
2. These variables cannot be local
3. The default values of static/class variables are the same as the Instance variables.
4. Static variables can be used within a program by calling the className.variableName
5. The memory allocated to store Static variables is Static memory

Arrays

Array Characteristics:

- Java arrays are dynamically allocated
- Arrays are objects, we can find their length using the object property length
- Array variable can also be declared like other variables with [] after the data type
- Variables in the array are ordered and each have an index beginning from 0
- Array can be also be used as a static field, a local variable or a method parameter
- Size of an array must be specified by an int or short value and not long
- The direct superclass of an array type is Object
- Every array type implements the interfaces Cloneable and java.io.Serializable
- Array can contain primitives (int, char, etc.) as well as object (or non-primitive) references of a class
- In case of primitive data types, the actual values are stored in contiguous memory locations
- In case of objects of a class, the actual objects are stored in heap segment.

UNICAL ACADEMY

Arrays

Single Dimensional Array
Multi Dimensional Array

Single Dimensional Array

Syntax to Declare an Array:

```
dataType[] arr; (or)
dataType []arr; (or)
dataType arr[];
```

Instantiation of an Array:

```
arrayRefVar=new datatype[size];
```

Multidimensional Array

Syntax to Declare Multidimensional Array:

```
dataType[][] arrayRefVar; (or)
dataType [][]arrayRefVar; (or)
dataType arrayRefVar[][]; (or)
dataType []arrayRefVar[];
```

Instantiation of Multidimensional Array:

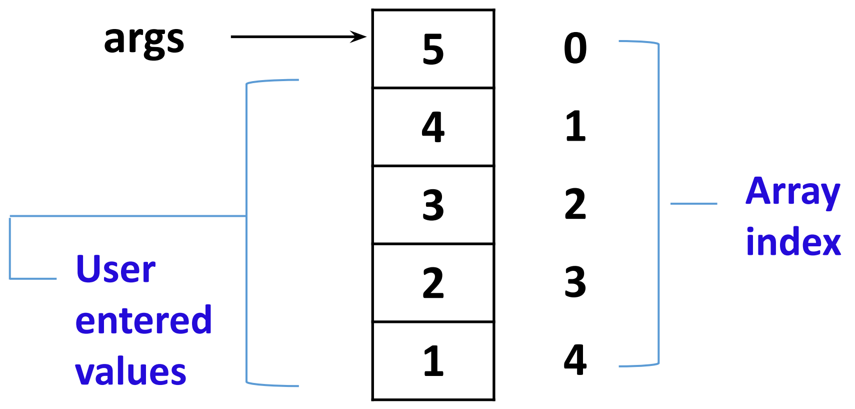
```
datatype[][] arr=new datatype [size][size];
```

Command-Line Arguments

When we invoke the application, the runtime system passes the command line arguments to the application's main method via an array of strings

```
public  static void main (String[] args)
```

Arguments are stored in the args array of the main method declared as –



If program needs to support a numeric command line argument, it must convert string argument that represents a number

```
int firstArg = 0;
        if (args.length>0) {
          firstArg = Integer.parseInt(args[0]);
        }
```

parseInt() method in the integer class throws a NumberFormatException (ERROR) if the format of args[0] is invalid

Session Recap

UNICAL ACADEMY