Learning Goals

This session will help you to understand,

- What are java keywords?

- What are java primitive data types?

- How to declare & define variables?

Keywords

- **_Keywords_** are reserved identifiers that are defined in java language.

- These words cannot be used by developers.

- Java will throw an compilation error when developer uses them in the program.

- Java keywords will be in lowercase letters and incorrect usage results in compilation errors.

Do

static

throws

Keyword Examples

abstract

Double

switch

public

private

Primitive Data Types

There are eight primitive data types

Data Type	Size In Bits	Range	Default Value
boolean	1	true or false	false
byte	8	-2^7 to $2^7 - 1$	0
short	16	-2^{15} to $2^{15} - 1$	0
int	32	-2^{31} to $2^{31} - 1$	0
long	64	-2^{63} to $2^{63} - 1$	oL
float	32	-2^{31} to $2^{31} - 1$	0.0f
double	64	-2^{63} to $2^{63} - 1$	0.0d
char	16	0 to $2^{15} - 1$	'\u0000'

Stores true/false

Used for storing whole numbers.

Used for storing decimal numbers.

Stores single character

Java Variables

Dictionary definition of a variable:

- A quantity that can assume any of a set of values.

- Something that is likely to vary; something that is subject to variation.

Let pick an analogy:

The box is used as a container to hold any thing. Be it fruits, toys etc..

What is a Java variable?

Similar to the box, Java variables are the unit of storage in a java program. It is used to store the state.

Variable Declaration

Syntax:

> **<data type> <variable name> [=initial value];**

Where,

- **Data Type** – Defines the type of value the variable can store. Mandatory.

- **Variable Name** – Name to identify the variable. Mandatory.

- **initial value –** Represents the initial value which needs to be assigned to a variable.

Example:

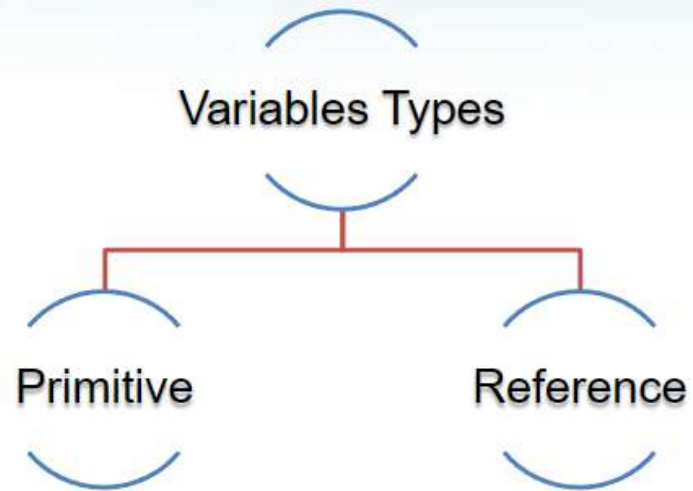> int age; **// Declared a variable age with data type int.**
>
> **(or)**
>
> int age =18; **// Age declared as int with default value 18**

Variables types in Java

Variables Types

Primitive

Reference

Example: int age;

Example: Student std;

This is a process of assigning an initial value to a variable.

How is it done?

<data Type> <variable Name> = <initialValue> ;

Illustration:
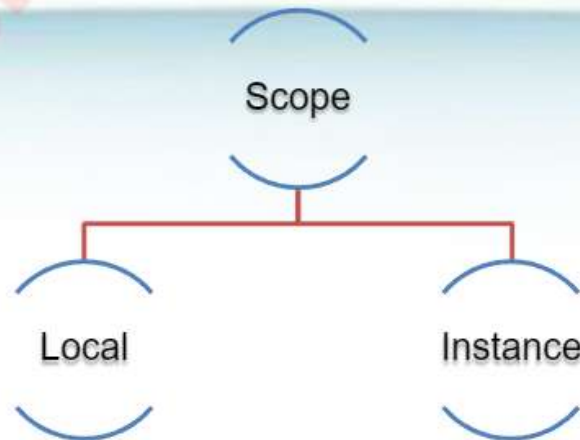
int age = 18;

The above statement declares a **variable** named **age** of data type **int** and assigns an initial **value** of **18** that can be changed.

Variables declaration scope

Scope

Local

Instance

- Variables which are declared inside a method.

- Can be accessed only inside the method in which they are declared.

- The state is lost once the method execution is complete.

- Variables are declared in the body of the class.

- The value of the variable is retained till the object state exist.

- This variable can be accessed by all the methods of the class.

```java
public class Actor {

    int age = 10;


    public void displaySalary()
    {
        int salary =1000;
        // logic to calculate and print salary

    }

}
```

Instance Variable – Declared inside the body of the class.

Local Variable – Declared inside the body of the method.

Variables Naming Convention

- The first character of a variable must be a letter, a dollar sign ($), or an underscore (_).

- But the ideal convention is to begin with a letter.

- Character **other than** the first character in an identifier may be a letter, a dollar sign, an underscore, or a digit.

    Example: price, $price, _price.

- Java language keywords (or reserved words) can be used as identifiers.

- If the variable name has only one word, then spell that word in all lowercase letters. If it consists of more than one word, then capitalize the first letter of each subsequent word.

    Example: employeeName, employeeAge

- Avoid abbreviations, the variable name should be meaningful.

    Don't: studentage, studentname.

    Do: studentAge, studentName.

What is a Java Literal?

A literal is a value assigned to a variable in a class.

- Literals appears on the right hand side of the variable declaration.

- Literals value can be assigned for any of the primitive data type.

Illustration:

int age = **50** ;

(or)

float price =**2000.0**;

A literal is a value and not a variable

Where "50" and "2000.0" are int and float literals respectively.

Types of literals

These are some literal examples.

Literal Type	Description	Formats and Example
Integer Literals	Integral literals may be represented by decimal, octal, or hexadecimal numbers. The octal number is prefixed with 0, and hexadecimal with 0x or 0X.	Decimal :12 Hexadecimal:0xC Octal:014
Floating Literals	A floating-point literal is represented by a floating-point number, and is used for both float and double data types.	Float:1234.5f Double:12.65,3.56E2 ,23d
Boolean Literals	A variable of boolean type can have only one of two possible values, true or false. Therefore, these two values are the only available boolean literals	true, false
Char Literals	A char literal may be represented by a single character enclosed in single quotes.	'a', '\u4567'
String Literals	String literals represent multiple characters and are enclosed by double quotes.	"Hello World"

1. Create a java class Actor with the following variables.

Variable Name	Data Type	Default value
actorAge	Int	22
actorSalary	Double	10,000
actorSavings	Float	8.0
actorPhone	Int	9885555

Solution

2. Create a method named calulateTax() which will instantiate the actor object and try printing the values of the instance variables, as follows,
    - "The age of the Actor is <actorAge> "
    - "The Salary of the actor is <actorSalary> "
    - "The actor phone number is <phone>"
    - "The total saving of actor is <actorSavings>".

3. Add a main method in the Actor class and invoke the calculateTax method and see if the values are displayed.

```java
public class Actor {

    int actorAge = 22;
    double actorSalary = 10000.0d;
    float tax = 8.0f;
    int actorPhone = 9885555;

    public void calculateTax() {
        Actor actor = new Actor();
        System.out.println("The age of the actor is" + actor.actorAge);
        System.out.println("The salary of the actor is" + actor.actorSalary);
        System.out.println("The tax of the actor is" + actor.tax);
        System.out.println("The phone number of the actor is" + actorPhone);
    }

}
```