## Features of Java:-

-Java is an object-oriented programming language.
-Java is open source.
-Java is Platform Independent
-Java is designed for the distributed environment of the internet.

## Identifier:-

-A name in java program is called Identifier. Which can be used for identification purpose.
-It can be method name, variable name, class name and label name.
-In Identifier allowed character-A-Z, a-z, $, _, 0-9
-Identifiers cannot start with Digit.
For eg :- int 123abc;      //Not allowed

int abc123;                //Allowed.
-Identifiers are case sensitive.

```
public class Test {             //Test is an identifier for class

    char ch = 'A';       //ch is an identifier for variable

    public void display() //display is an identifier for method
    {
        int x = 10;
    }
}
```

## Reserved Word:-

-In java some words are reserved words to represent the meaning of functionality such types of words are called reserved words.

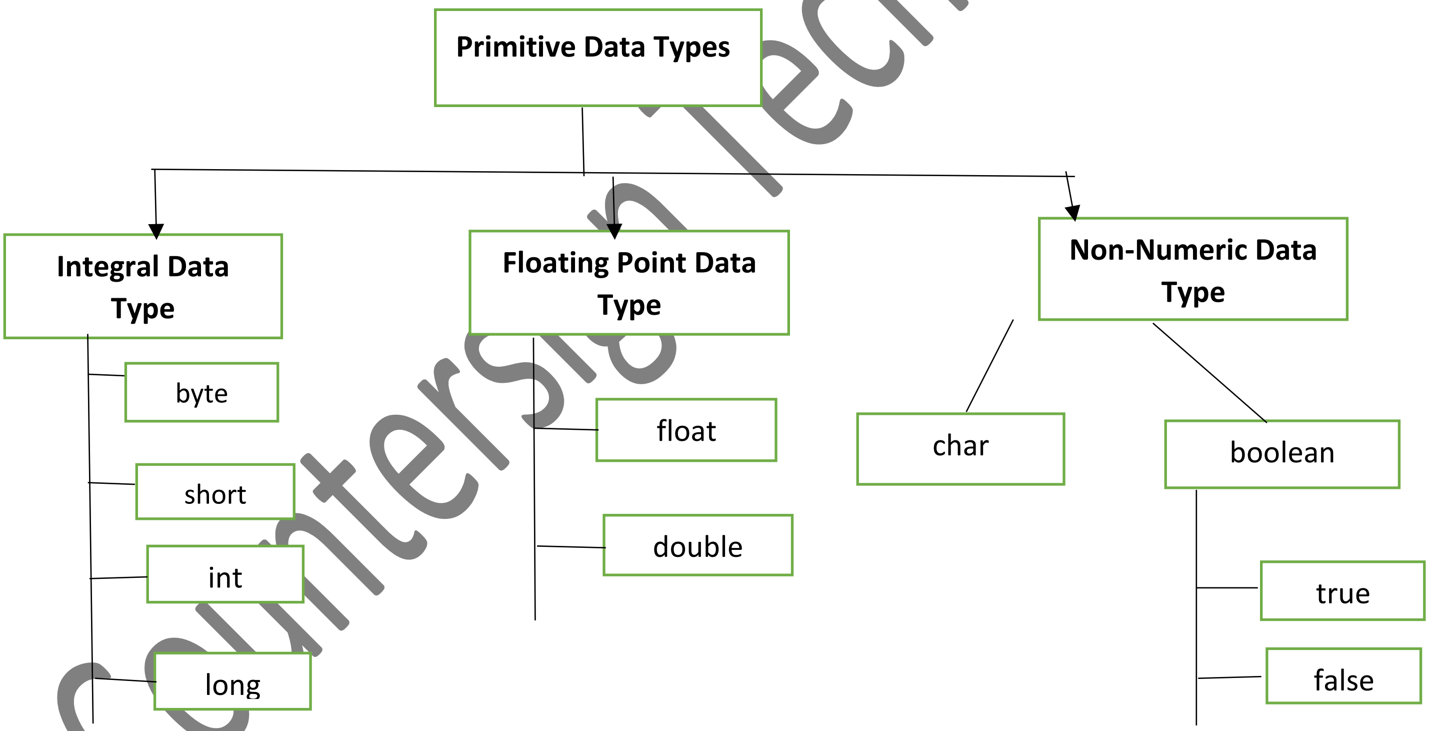
There are two types of reserved keywords: -

1. Keyword: - There is 50 Keywords in java like if ,else ,do ,while ,int ,float etc.
2. Reserved Literal: - true ,false ,null

## Data Types

-In java every variables and expression has some type, each and every data type is clearly define.
-Data types specify the different sizes and values that can be stored in the variable.
-There are two types of data types in Java:

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



## Integral Data Types: -

### Byte:

- Bytes is 8-bit long (1 byte).

-Max value =127

-Min value = -128

For eg: - byte b=125 //Allowed

byte b=133 //Not Allowed. Exceeds range. Loss of data may happen

### Short:

-Short is 2-byte long (16 bits).

-Max size = 32767 ($2^{15}$-1)

-Min Size= -32768 (-$2^{15}$)

### Int:

-Int is 4-bytes long (32 bits).

-Most commonly used.

-Max Size= 2,147,483,647 ($2^{31}$-1)

-Min Size= -2,147,483,646 (-$2^{31}$)

Ex. int a=10, b=5; //*Allowed*

### Long:

-Long is 8-byte long (64 bits).

-Max Size= 9,223,372,036,854,775,807 ($2^{63}$-1)

-Min Size= -9,223,372,036,854,775,806 (-$2^{63}$)

-It is used where int is not sufficient to store data.

Ex. Calculate number of seconds in 10 years. (10x365x24x60x60)

## Floating Point Data type: -

**1. Float:-**

-float data type is used for 5 to 6 decimal places of accuracy.
-Size-4 byte
-Range -3.4e38 to +3.4e38
Ex. float p= 5.18f *//append f at the last*

**2. double: -**

-If we want 14 to 15 decimal places of accuracy, then we should go for double.
-Size-8 byte
-Range: -1.7e38 to 1.7e38
Ex. Double d=9.3254768;

## Non-Numeric Data Type:-

**char: -**

Size- 2 byte
- Java uses *Unicode* to represent characters.
- It's range is 0 to 65,536
Ex. char ch = 'S' *//Value should be declared in single quotes*

**boolean: -**

-Size - virtual machine dependent.
-It save only two values true and false.
Ex. boolean b=true;
boolean b = false;

## Non – Primitive Data Type: -

1. String
- String is basically an object that represents sequence of char values
2. Array
- Array is an indexed collection of fixed number of homogenous data element.
3. Class

- A class is group of member function (methods)variables data, printing statement, scanning statement, Object, Condition statement, loops, etc.
- A class can be defined behavior, state of that object of its type support
- We can create Class under package. In one package we can create multiple classes.

## Variables: -

-Variables are used to store data.
-Variables are combination of data type, identifier and optional initialization.
-Based on position of declaration and behavior of all variables are divided into three types:-

1. Global Variable
2. Static Variable
3. Local Variable

**Global Variable: -**

-Global variable should be declare with in the class directly but outside of any method or block or constructor.
-Global variable will be created at the time of object creation and discard at the time of object discard.
-Global variable also known as Instance variable or object level variable.

**Local Variable: -**

-To meet temporary requirement of programmer we can declare variable inside a method or block or constructor such types of variables are called Local Variables.

-The only allowed modifiers for local variable are '*default*' and '*final*'. Public, private, protected, static are not allowed.

-Local variable's scope is limited to the method, block or constructor inside which it is defined

For ex.

```
          public class Test
               {
                 static int x=11; //Static Variable/Global Variable
                  int y=10;  //Global Variable /Instance Variable
              public void m1()
                  {
                   int z=20;  //Local variable
                  }
             Public static void main(String[] args)
                 {
                           Test t1=new Test();
                          System.out.println(x); //11
                          System.out.println(t1.y); //10
                          System.out.println(Test.x); //11
                          System.out.println(z); //Compile Time Exception
                 }
}
```

## Keywords: -

- Keywords are predefined, reserved words used in Java programming that have special meanings to the compile. There are some used reserved keywords and some are the unused Keyword.
- goto and const are the unused keyword.
- Below is the reserved used keyword

| No. | Keyword | Description |
|---|---|---|
| 1. | **abstract** | Specifies that a class or method will be implemented later, in a subclass |
| 2. | **assert** | Assert describes a predicate placed in a java program to indicate that the developer thinks that the predicate is always true at that place. |
| 3. | **boolean** | A data type that can hold True and False values only |
| 4. | **break** | A control statement for breaking out of loops. |
| 5. | **byte** | A data type that can hold 8-bit data values |
| 6. | **case** | Used in switch statements to mark blocks of text |
| 7. | **catch** | Catches exceptions generated by try statements |
| 8. | **char** | A data type that can hold unsigned 16-bit Unicode characters |
| 9. | **class** | Declares a new class |
| 10. | **continue** | Sends control back outside a loop |
| 11. | **default** | Specifies the default block of code in a switch statement |
| 12. | **do** | Starts a do-while loop |
| 13. | **double** | A data type that can hold 64-bit floating-point numbers |
| 14. | **else** | Indicates alternative branches in an if statement |
| 15. | **enum** | A Java keyword is used to declare an enumerated type. Enumerations extend the base class. |
| 16. | **extends** | Indicates that a class is derived from another class or interface |
| 17. | **final** | Indicates that a variable holds a constant value or that a method will not be overridden |

| | | |
|---|---|---|
| 18. | **finally** | Indicates a block of code in a try-catch structure that will always be executed |
| 19. | **float** | A data type that holds a 32-bit floating-point number |
| 20. | **for** | Used to start a for loop |
| 21. | **if** | Tests a true/false expression and branches accordingly |
| 22. | **implements** | Specifies that a class implements an interface |
| 23. | **import** | References other classes |
| 24. | **instanceof** | Indicates whether an object is an instance of a specific class or implements an interface |
| 25. | **int** | A data type that can hold a 32-bit signed integer |
| 26. | **interface** | Declares an interface |
| 27. | **long** | A data type that holds a 64-bit integer |
| 28. | **native** | Specifies that a method is implemented with native (platform-specific) code |
| 29. | **new** | Creates new objects |
| 30. | **null** | This indicates that a reference does not refer to anything |
| 31. | **package** | Declares a Java package |
| 32. | **private** | An access specifier indicating that a method or variable may be accessed only in the class it’s declared in |
| 33. | **protected** | An access specifier indicating that a method or variable may only be accessed in the class it’s declared in (or a subclass of the class it’s declared in or other classes in the same package) |
| 34. | **public** | An access specifier used for classes, interfaces, methods, and variables indicating that an item is accessible throughout the application (or where the class that defines it is accessible) |
| 35. | **return** | Sends control and possibly a return value back from a called method |
| 36. | **short** | A data type that can hold a 16-bit integer |

| | | |
|---|---|---|
| 37 | **static** | Indicates that a variable or method is a class method (rather than being limited to one particular object) |
| 38. | **strictfp** | A Java keyword is used to restrict the precision and rounding of floating-point calculations to ensure portability. |
| 39. | **super** | Refers to a class's base class (used in a method or class constructor) |
| 40. | **switch** | A statement that executes code based on a test value |
| 41. | **synchronized** | Specifies critical sections or methods in multithreaded code |
| 42. | **this** | Refers to the current object in a method or constructor |
| 43. | **throw** | Creates an exception |
| 44. | **throws** | Indicates what exceptions may be thrown by a method |
| 45. | **transient** | Specifies that a variable is not part of an object's persistent state |
| 46. | **try** | Starts a block of code that will be tested for exceptions |
| 47. | **void** | Specifies that a method does not have a return value |
| 48. | **volatile** | This indicates that a variable may change asynchronously |
| 49. | **while** | Starts a while loop |