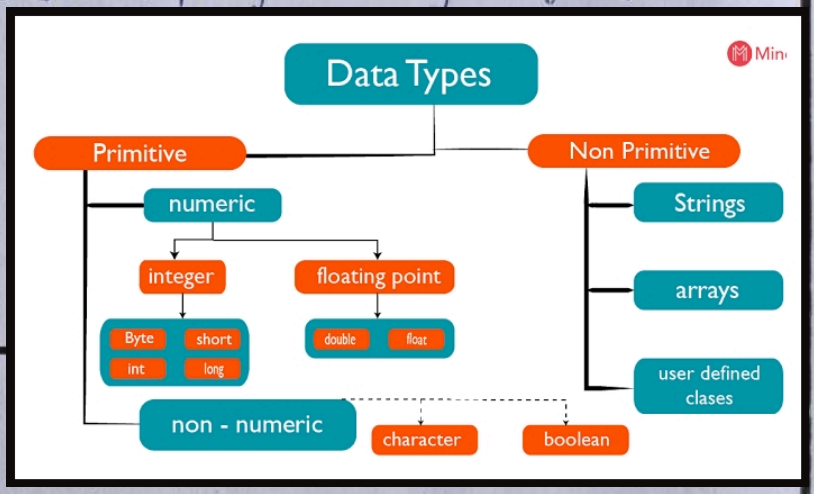
INTRODUCTION JAVA

Java is a programming language and a platform. Java is a high level robust, object-oriented and secure programming language.



JAVA EXAMPLE:

```
Class Example {
  Public static void main (string args [])
  {
    system.out.Println ("Hello Java");
  }
}
```

APPLICATION :-

a). Standalone / Desktop application
b). Web Application.
c). Enterprise Application.
d). Mobile Application.

JVM :- JVM stands form Java Virtual machine is an abstract machine. It is called a virtual machine because it doesn't physically exist. JVM is used to provide a runtime environment in which Java bytecode can be execute.

JRE :- JRE stands form Java Runtime Environment. The Java Runtime Environment is a set of softwere tools which are used for developing Java applications. It is used to provide the runtime environment. It is the implementation of JVM. It is physically exist.

JDK :- JDK stands form Java Development Kit. The Java Development Kit (JDK) is a software Development environment which is used to develop Java applications and applets. It is physically exist. It contains JRE + Development tools.

JAVA VARIABLES : A Variable is a container which holds the value while Java Program is executed. A variable is assigned with data type.

```
int data = 50;
```

TYPES OF VARIABLES

There are three types of variables in Java :

- local variable
- instance variable
- static variable.

1). **Local Variable :** A Variable declared inside the body of the method is called the local variable.

2). **Instance variable :** A variable declared inside the class but outside of the method is called the instance variable.

3). **Static variable :** A variable that is declared as static is called a static variable. It cannot be local. you can create a single copy of the static variable and share it among all the instance of the class. Memory allocation for static variables happens only once when the class is loaded in the memory.

JAVA CONTROL STATEMENTS

Java If-else statement : The Java if statement is used to test the condition. It executes the if block of the condition is true otherwise else block is executed.

Syntax:

```
if ( condition) {
    // code if condition is true
} else {
    // code if condition is false.
}
```

Java if-else-if ladder statement : The if-else-if statement executes one condition from multiple statements.

syntax:

```
if (condition) {
  // code if condition 1 is true.
else if (condition 2) {
  // code to be executed if condition 2 is true }
else if (condition 3) {
  // code to be executed if condition 3 is true }
else {
  // code to be executed if all the condition
     are false.
}
```

Java switch statement :

The Java switch statement executes one statement from multiple conditions.

In the other words, the switch statement tests the equality of a variable against multiple values.

Syntax :

```
switch (expression)
{
    case value1: // code to be executed;
                 break;
    case value 2: // code to be executed;
                  break;
    case value 3: // code to be executed;
                  break;
    default: // code to be executed if all cases are
             not matched;
}
```

Example :

```
switch (number) {
case 10 : system.out. println ["10"];
          break;
case 20 : system.out. println ("20");
          break;
case 30 : system.out.println ("30");
          break;
default : system.out. println ("NO.1 in 10, 20, or 30");
}
```

LOOPS IN JAVA

Java simple for loop : A simple for loop is the same as C/C++. We can initialize the variable, check condition and increment/decrement value.

Syntax :

```
for(initialization ; condition ; increment/Decrement)
{ // statement or code to be executed
}
```

Java Nested for loop : If we have a for loop inside the another loop, it is known as nested for loop.

Example :

```
for(int i = 1 ; i<=3; i++) {
for(int j = 1 ; j<=3; j++) {
  system.out.println(i+" "+j);
} // end of i
} // end of J
```

Java for each loop : For each loop is used to traverse array or collection in Java.

Syntax :

```
for(data-Type variable : array.Name) {
    // code to be executed
}
```

JAVA ARRAY

Array : An array is a collection of multiple data which are having similar data type. Array in Java is Index-based, the first element of the array is stored at the 0th index, 2nd element is stored on 1st Index and so on.

Advantages :

- Code Optimization : It makes the code optimized. We can retrieve or sort data.
- Random access : We can get only data located all an Index position.

Disadvantages :

- size limit : We can store only the fixed size elements in array. It doesn't grow its size at runtime.

Syntax to Declare an array In Java

```
data type[] arr ;
data type arr[] ;
```
or

Instantiation of an Array in Java

```
arrayRefval = new datatype(size);
```

Types of Array in Java

There are two types of array

1. Single Dimensional Array
2. Multi Dimensional Array

1. **Single Dimensional Array :** An array with only one subscript that array is known as single / one Dimensional Array.

Example:

```
int a[] = new int[5];
```

2. **Multi Dimensional Array :** An array with more than or two subscript that is called as Multi Dimensional Array. Mostly Multi Dimensional Array is used for table representation.

Example:

```
int a[][] = new int[5][5];
```

3. **Anonymous Array in Java :** Java supports the feature of an anonymous array, so you don't need to declare the array while passing an array to method.

Example : `printArray(new {10,20,30,40});`