

1. In Java applications, it is suggestible to provide identifiers with a particular meaning.

```
String xxx = "abc123"; ---> Not Suggestible  
String accNo = "abc123";---> Suggestible
```

2. In Java applications, no length restriction for the identifiers , we can write identifiers with any length, but, it is suggestible to provide identifiers with around 10 symbols

```
String temporaryemployeeaddress = "Hyd"; --> Not Suggestible  
String tempEmpAddr = "Hyd"; ----> Suggestible
```

3. If we have multiple words in single identifier then it is suggestible to separate multiple words with special notations like '_' symbols.

EX:

```
String tempEmpAddr = "Hyd"; -----> Not Suggestible  
String temp_Emp_Addr = "Hyd";----> Suggestible
```

To prepare Java applications, JAVA has provided a set of Literals.

1. Integral / Integer Literal Group:

byte, short, int, long ---> 10, 20,.....
char -----> 'A', 'b', 'c', '\n', '\t',...

2. Floating Point Literals:

float -----> 22.22f, 234.345f
double -----> 345.3456D, 678.456d, 3456.1234

3. Boolean Literals:

boolean -----> true, false

4. String Literals:

String ----> "abc", "def",

Number Systems:

In all PLs, to represent numbers we must use some systems called as Number Systems.

1. Binary Number System[BASE-2]
2. Octal Number System[BASE-8]
3. Decimal Number System[BASE-10]
4. Hexa Decimal Number System[BASE-16]

Number Systems:

In all PLs, to represent numbers we must use some systems called as Number Systems.

1. Binary Number System[BASE-2]
2. Octal Number System[BASE-8]
3. Decimal Number System[BASE-10]
4. Hexa Decimal Number System[BASE-16]

```
1 class Test
2 {
3     public static void main(String[] args)
4     {
5         int a = 0B1010;
6         System.out.println(a);
7     }
8 }
```

1. Binary Number System[BASE-2]:

Alphabet : 0, 1
Prefix : 0b or 0B

EX:

int a = 10; ----> It is not binary number, it is decimal number
int b = 0b1010; ---> Valid
int c = 0B1100; ---> Valid
int d = 01101; ----> Invalid
int e = 0B1020; ---> IN|

2. Octal Number System[BASE-8]:

Alphabet: 0,1,2,3,4,5,6 and 7
Prefix : 0[Zero]

int a = 10; -----> It is not Octal number, it is decimal number
int b = 0123; ---> valid
int c = 03456; --> Invalid
int d = 05678; --> Invalid

3. Decimal Number System[BASE-10]

Alphabet: 0,1,2,3,4,5,6,7,8,9
Prefix : No prefix.

4. Hexa Decimal Number System[BASE-16]

Alphabet: 0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f
Prefix : 0x or 0X|

```
int a = 10; -----> It is not Hexa Decimal, it is decimal number
int b = 0x123456; ---> Valid
int c = 0X789abcd; ---> Valid
int d = 0x123defg; ---> Invalid
```

```
1 class Test
2 {
3     public static void main(String[] args)
4     {
5         int a = 0b10;
6         System.out.println(a);
7         int b = 010;
8         System.out.println(b);
9         int c = 0x10;
10        System.out.println(c);
11    }
12 }
```

```
1 class Test
2 {
3     public static void main(String[] args)
4     {
5         int a = 0b1010;
6         int b = 0b10;
7         System.out.println(a+b);
8         System.out.println(a-b);
9         System.out.println(a*b);
10    }
11 }
```

```
1 class Test
2 {
3     public static void main(String[] args)
4     {
5         int a = 01010;    I
6         int b = 010;
7         System.out.println(a+b);
8         System.out.println(a-b);
9         System.out.println(a*b);
10    }
11 }
```

```
1 class Test
2 {
3     public static void main(String[] args)
4     {
5         int a = 0x1010;
6         int b = 0x10;
7         System.out.println(a+b);
8         System.out.println(a-b);
9         System.out.println(a*b);
10    }
11 }
```

```
1 class Test
2 {
3     public static void main(String[] args)
4     {
5         int a = 0b1010 + 010;
6         System.out.println(a);
7     }
8 }
```