Language Fundamentals

1. Identifiers
2. Reserved words
3. Data Types
4. Literals
5. Arrays
6. Types of variables
7. Var - arg methods
8. main method
9. command line arguments
10. Java coding standards

IDENTIFIERS → which can be use for Identification purpose.

Eg → class name, method name, variable name & label name (l_1, l_2).

A name in a java program is called a identifier which can be used for identification purpose.

Eg → class Test ①{
     public static void main ②(string ③
        [] array ④)
    {
       int x ⑤ = 10;
    }
}

In this example, 5 identifiers are there.

* Rules for defining Identifiers

① only allowed identifiers —
a to z, A to Z, 0 to 9, $ & underscore.

eg → total_number (✓)
total # (✗)

If we use any other charcater
we will get compile time
error.

② total123 (✓)
123total (✗)

Identifiers should not start with
digits.

③ Java identifiers are case-sensitive.
Java language itself is a case-
sensitive language.

```
class Test
{
    int number = 10;
    int Number = 20;
    int NUMBER = 30;
}
```

④ There is no length limit for JAVA identifiers but it is not recommended to take too lengthy identifiers.

⑤ We can't use reserved words as identifiers.

eg → int x = 10; (✓)
     int if = 20; (✗) Invalid

⑥ All pre-defined java class names & interface names can be use as identifiers.

eg → class Test {
         public static void main (String [] args)
         {
             int String = 888; | int Runnable = 999;
             Super (String); | Super (Runnable);
         }
     }

Given though it is valid but it is not good programming practice because it reduces readability & create confusion.

Ques : which of the following are valid
JAVA identifiers ?

total_number (✓)
total.# (✗)
123total (✗)
total123 (✓)
ca$h (✓)
_$_$_$_$ (✓)
all@hands (✗)
Java2share (✓)
Integer (✓)
Int (✓)
int (✗)