```
/*
==================================
Data type and Types of data types
==================================
01. Need of this chapter?
        - In this chapter we will learn 'different data types' available in Java
          for creating 'variables and objects' for 'storing single value' and
          'multiple values' as group in a program.

        - For storing a value in a program we must allocate memory

        - To to tell to Compiler and JVM for allocating memory,
          we must use a set of keywords called data types

02. What is a data type?
        - A 'keyword or a class name' that is used for creating
          a variable or an object memory for storing 'single value' or
          multiple values' as one group in the program is called data type.

03. Why data type?
        - A data type is used for allocating variable or object memory
          for storing values in the program.

04. What is the information provided by the data type to the compiler and JVM?
        - A data type (keyword or classname) provides below information
                1. The type of memory (Integer/Floating-
point/Character/Boolean/String)
                2. The size of the memory (1 or 2  or 4 or 8 bytes)
                3. The type of the value and the range of the value can be stored
                4. The allowed operators to apply on this variable
                5. The result type comes out from an expression

        - Whichever the keyword r classname provides above information
          that keyword is called data type

          For example:
                - int, double, char, boolean are data type keywords
                - public, void, if, ... are not  data type keywords

05. Can we store value in a program direclty?
        No, it leads CE: not a statement

06. Can we storing a value in a variable without creating it?
        No, it leads CE: can not find symbol variable

07 Below program explains
        1. creating variable
        2. storing value in the variable
        3. reading variable value and printing
        4. printing variable name and its value by using + operator
        5. diff compile time errors occured in this program
*/
class Test01_StroingData {

        public static void main(String[] args) {
                //Data type keywords -> [b, s, i, l], [f, d], [c], [bo]
                //class name         -> [String]

                //10;           //storing value in the program direclty -> not allowed
-> wrong syntax
```

```
                //i = 10;       //storing value in a variable -> not alloWed ->
variable is not yet created

                int i ;         //creating new variable of type int with the name i
                i = 10;         //storing value 10 in the variable i

                //10.5;         //CE: not a statement, we can not store value directly
in the program
                //d = 10.5;     //CE: can not find symbol variabled, we did not create
variable d

                double d;       //creating new variable d of type double for storing
floating-point type value
                d = 10.5;       //storng value 10.5 in the variable d

                char ch;        //creating new variable ch of type char for storing
single character
                ch = 'a';   //storing character 'a' in the variable ch

                boolean bo;     //creating new variable bo of type boolean for storing
boolean value true/false
                bo = true;      //storing booelan value true in the variable bo

                //hari;                 //CE: not a statement
                //s1 = hari;    //CE: can not find symbol variable s1

                String s1 ;     //creating a variable s1 of type String for storing
string data (mulitple characters)
                //s1 = hari;  //CE: cannot find symbol variable hari
                s1 = "hari";    //for considering sequence of characters as string
                                                //we must place them inside "", else
it is considered as variable name

                System.out.println(i);  //accessing variable i for reading its value
and printing
                System.out.println(d);  //accessing variable d for reading its value
and printing
                System.out.println(ch); //accessing variable ch for reading its value
and printing
                System.out.println(bo); //accessing variable bo for reading its value
and printing
                System.out.println(s1); //accessing variable s1 for reading its object
and printing

        }
}
```