DAY 4- Java

Indentifiers: Identifiers are the valid names given to the different fields in java file, those fields may be variables, methods, class etc.
Identifiers should follow the below mentioned rules:

1. Should not start with number.
2. Can contain a-z, A-Z, 0-9, underscore(_), dollar ($).
3. Keywords should not be used as identifier names.
4. Case-sentsitive (**name, Name** are considered as two different variable names)

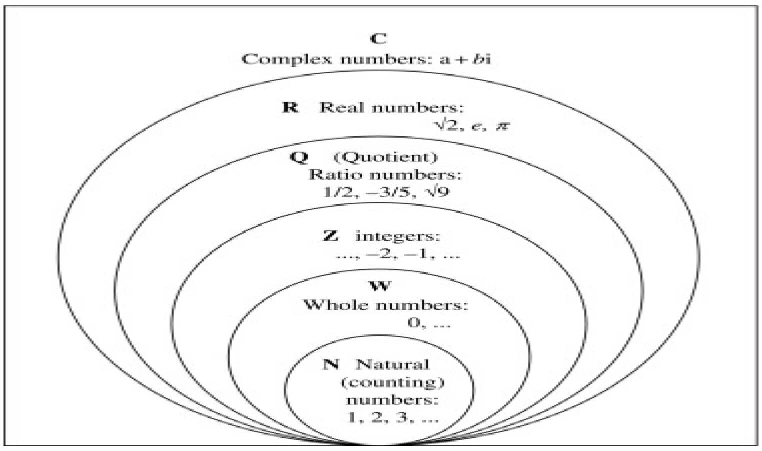
Constants: Constants are the fixed values, once assigned they can't be modified. In java the constant values are achieved using ***final*** keyword.
Ex: final float PI = 3.14;
Ex: final float GRAVITY = 9.81;

Variables: Variables are the named containers that are used to store data.
Ex: int x=10;
Ex: char c='b';
x is a variable that is storing integer value 10.
c is a variable that is storing character value 'b'.

print statement:
*System.out.println()***:** is used to display the outputs or some user friendly messages.
System.out.print(): after printing the output, the pointer will continue in the same line for displaying any other data.
System.out.println(): after printing the output, the pointer will move the next line and the other data will be displayed in the new line.

Java is a statically typed language:
In java we usually use data types to mentioned the type of data we are using. The advantage of static type is that we can restrict the data that is being used in the program. So, errors of type mismatched are caught at compile time reducing run time errors.
Ex: You are asked to enter the age, so age will definitely be a number, so it will restrict the user if he try to enter any other kind of data.
int number = 10; // 'number' must be an integer
number = "Hello"; // Error: Type mismatch (String cannot be assigned to an int)

Datatypes:
Primitive: the basic form of any entity.
Ex: For fire, the primitives are stones
EX: For name, the primitives are characters



Primitive datatypes:

1. **Numbers:**
   Complete number: *byte(1 byte), short (2 bytes), int (4 bytes), long (8 bytes)*
   Decimal number: *float(4 bytes), double(8 bytes)*
   Note1: Different datatypes occupy different size in memory. The need of having four different number datatypes are because of the use cases like:
   If you want to store human age in a variable it better to choose byte datatype because it will occupy only 1 byte of memory, using any datatype like short, int, long will occupy more space, which is simply waste of memory.
   Assume, for storing a person's age it will cose 1 rupee for using byte datatype, where as it may cost 8 rupees if we use long type.
   Note2: floating will have lower precision where as double have higher precision.
2. **Character:** stores character values (a-z, A-Z, special symbols)
   char c='a'
3. **Boolean:** stores true or false
   boolean value=true;

Non-Primitive: Other than primitive, everything that are used to store data are called non-primitive.
Ex: String, Array, Collection frameworks.

Scanner class:

Scanner is a class from util package that is used to take input from the user.
Scanner scan=new Scanner(System.in);
scan is the object of class Scanner, that is used to take various inputs from its methods.

for integer data:
byte b1=scan.nextByte();
short s1=scan.nextShort();
int i1=scan.nextInt();
long l1=scan.nextLong();
for floating point input:
float f = sc.nextFloat();
double d = sc.nextDouble();

for string input:
String str=scan.next(); //it will read a single word
String str=scan.nextLine(); //it will read complete sentence until enter is pressed.
for character input:
char c=scan.next().charAt(0);
for boolean input:
Boolean bool=scan.nextBoolean();

Case-Sensitive: C, C++, Java, Python, Kotlin, JavaScript, Rust, Go, Swift.
Case-Insensitive: SQL (mostly), HTML, BASIC, Pascal