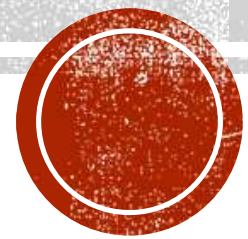
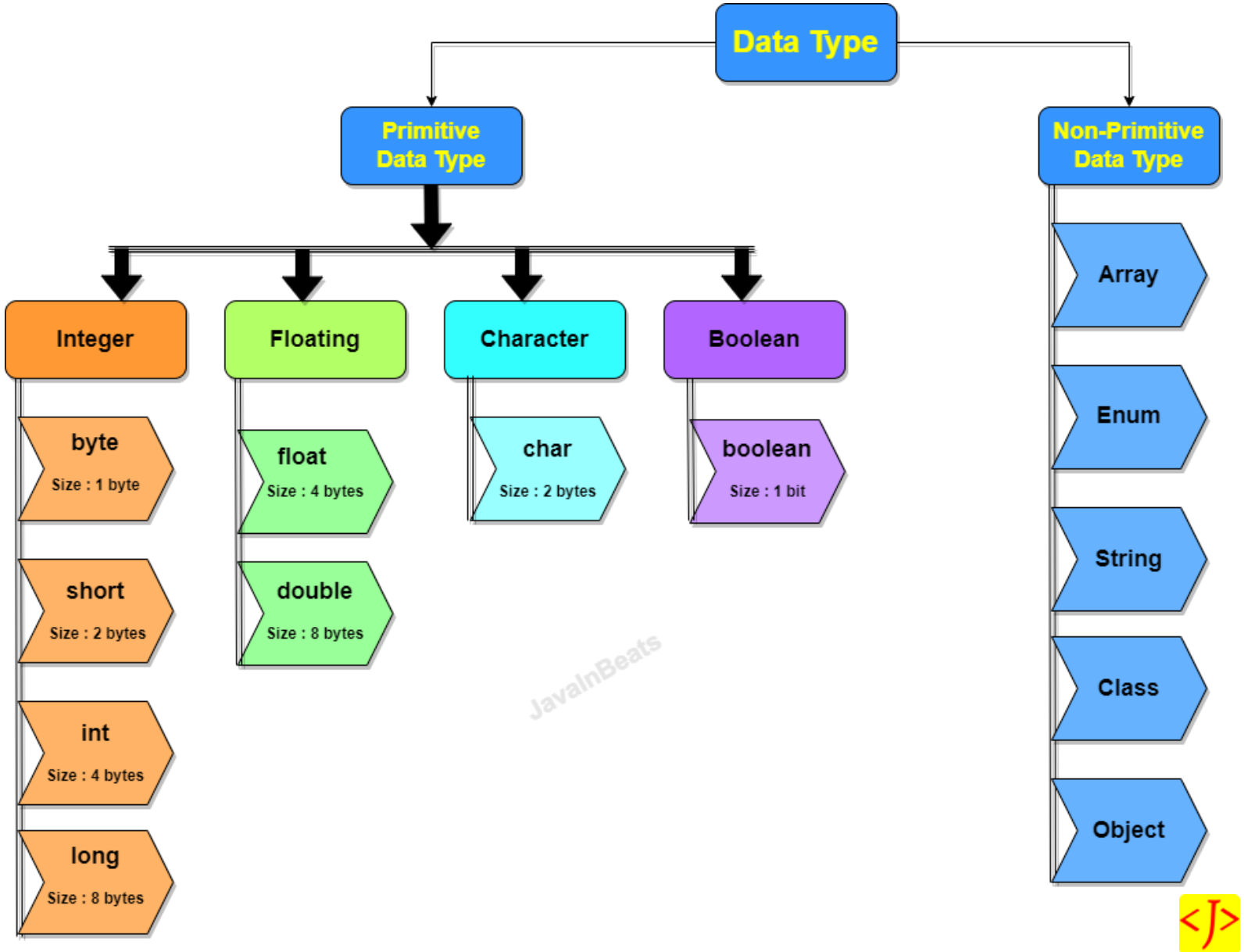
DATA TYPE AND VARIABLE



WHY DATA TYPE

1. Data Types are required to specify the type of values used in a program.
2. Data types are used to create a variable.
3. There are 2 types of data type in java
    a. Primitive Data Type
        i. Primitive Data type has a fixed size which is predefine in java.
        ii. Primitive data type are in non-object format.
    b. Non-Primitive Data Type
        i. Non-primitive data type doesn't have any fixed size, its size will be decided as per it uses in the code.
        ii. Non primitive data types are always in the Object format.

CLASSIFICATION OF DATA TYPE



PRIMITIVE DATA TYPE

- **Integer**
  1. The numeric type of data which is without any decimal number.
  2. The values can be negative or positive.
  3. Any non-decimal numeric value created without any data type is consider as int.
  4. Example: 121, 10, -20
- **Floating**
  1. The numeric value with or without decimal number.
  2. The values can be negative or positive
  3. Any decimal numeric value created without any data type is consider as double.
  4. Example: 22332.12, 4.2, -121.1
- **Character**
  1. Can store a alphabets, numbers or symbols.
  2. There can be maximum one alphabet store inside char.
  3. Can store numeric value which must be non-decimal and only positive value allowed.
  4. You can store a ASCII values inside this.
  5. Character value has to store in a single quotes only if it is alphabet or symbol
- **Boolean**
  1. In Boolean you can store only true or false value.
  2. In java 0 and 1 in not consider as false and true

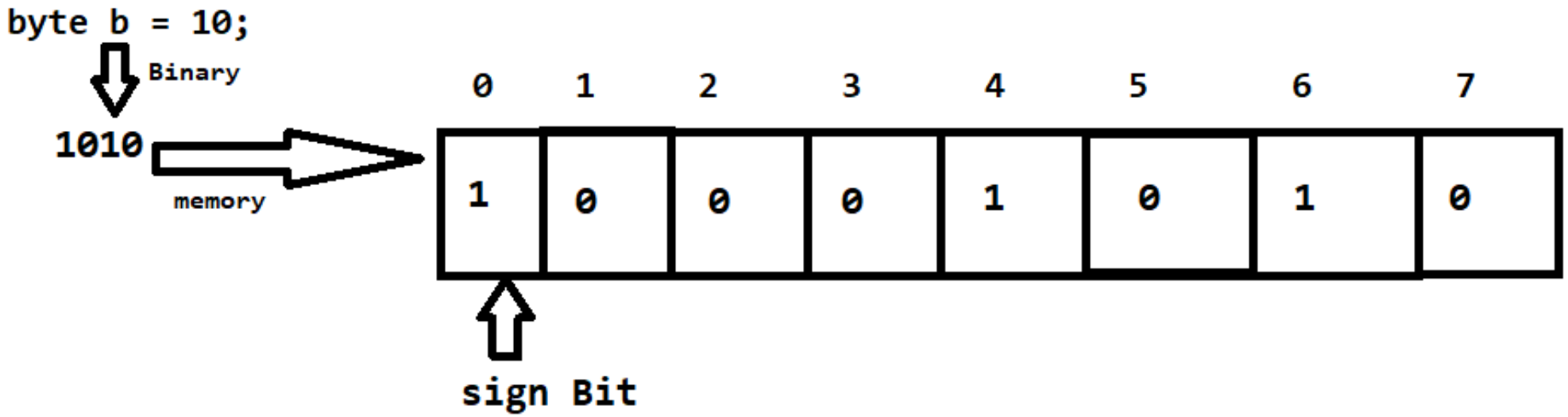
VARIABLE

- **Use of Variable**
  - Holding unique attribute data for an instance variable.
  - Assigning the value of one variable to another.
  - Representing values within a mathematical expression.
  - Printing value on screen.
  - Holding Reference to other objects.

- **Variable Declaration and initialization**
  - Syntax for attribute variable declaration and initialization.
    [access_modifier] DataType identifier [=value]
  - Syntax for declaring variable
    DataType identifier;
  - Syntax for initialization variable
    identifier = value;
  - Syntax for declaration and initialization
    DataType identifier [=value];

VARIABLE INSIDE MEMORY

- **Memory Foot prints for the variable**



- **Formula to calculate the Range of the Numeric values**

2^{n-1} to $2^{n-1}-1$

Here, the n is the number of bits

Example

byte : 2^{8-1} to $2^{8-1}-1$ => -128 to 127

RULES TO USE VARIABLE

- The numeric non-decimal values created without any data type by default allocated with the int memory.
- The numeric decimal values created without any data type by default allocated with the double memory.
- long
  a. Long values should be created with suffix L or l.
  b. Example: long phoneNumber = 9988770066L;
- float
  a. float values must be created with the suffix F or f.
  b. Example : float gravity = 9.8F;
- char
  a. You can store single character or symbol as a values. This value must be in a single quotes (‘ ’)
  b. You can also store a numeric non-decimal value inside char. It must be 0 or positive value.
  c. Every char values can be represent as per a ASCII table.
     - https://www.asciitable.com/
  a. Example:
     - char c1 = ‘A’;
     - char c2 = ‘@’;
     - char c3 = 65;
     - char c4 = -65; // not allowed
- boolean
  a. Boolean value must be **true** or **false** only.
  b. In java 0 and 1 is not consider as a Boolean value.
  c. Example: boolean b = true;
     boolean b = 1; // not allowed

CASTING IN PRIMITIVE DATA TYPE

- Casting is a process in which you can convert the value from one data type to another.
- There are 2 types of casting.
  - **Implicit Casting**
    i. The conversion of the data type from one type to another happens internally.
    ii. Example:
      - **byte a = 10;**
      - **int b = a; // Implicit casting**

  - **Explicit Casting**
    i. The conversion from one data type to another will have to do manually else compiler will give an error at compilation.
    ii. Explicit casting can give you logically incorrect output if you handled it inappropriately.
    iii. Example:
      - **int x = 10;**
      - **byte y = (byte) x; // explicit casting**

TYPE OF VARIABLE

1. The type of variable is decided on the bases of position of the variable created inside program.
2. There are 3 types of variables
   a. **Local Variable**
      i. The variables which are created inside a method is called as local variable.
      ii. This variable can be access inside a method only. It cannot be access outside method.
      iii. Local variables have to initialized before use else compiler will give an error.
   b. **Instance variable**
      i. The variable which is created inside class and outside any method is called instance variable.
      ii. These variables can be access in all the method of the class which is non static. These variables cannot be access in static method.
      iii. If you do not initialized the instance variable then java will initialized it with default value when the class object gets created.
   c. **Static/class Variable**
      i. The variables which are created inside class and outside any method using static keyword is called static or class variable.
      ii. This variables can be access in all the method of the class.
      iii. If you don't initialized this variable then it will get initialize by default values at class loading.

DEFAULT VALUE

Data Type	Value
byte	0
short	0
int	0
long	0L
float	0.0F
double	0.0D
char	\u0000
boolean	false
Non-primitive	null

Note: The Default values are only appliable for the Instance and Static variable

TASK

- **Task-1**

  WAP to store the product value and print the value of the product.
  Store the Product id, product price, product quantity
  And print the product id, product price, product quantity and the total product price (price * quantity)

- **Task-2**

  Calculate electricity bill, consider per unit price is 3.5 rupee and tax is 10 percent. Print the payable amount for given consume unit.