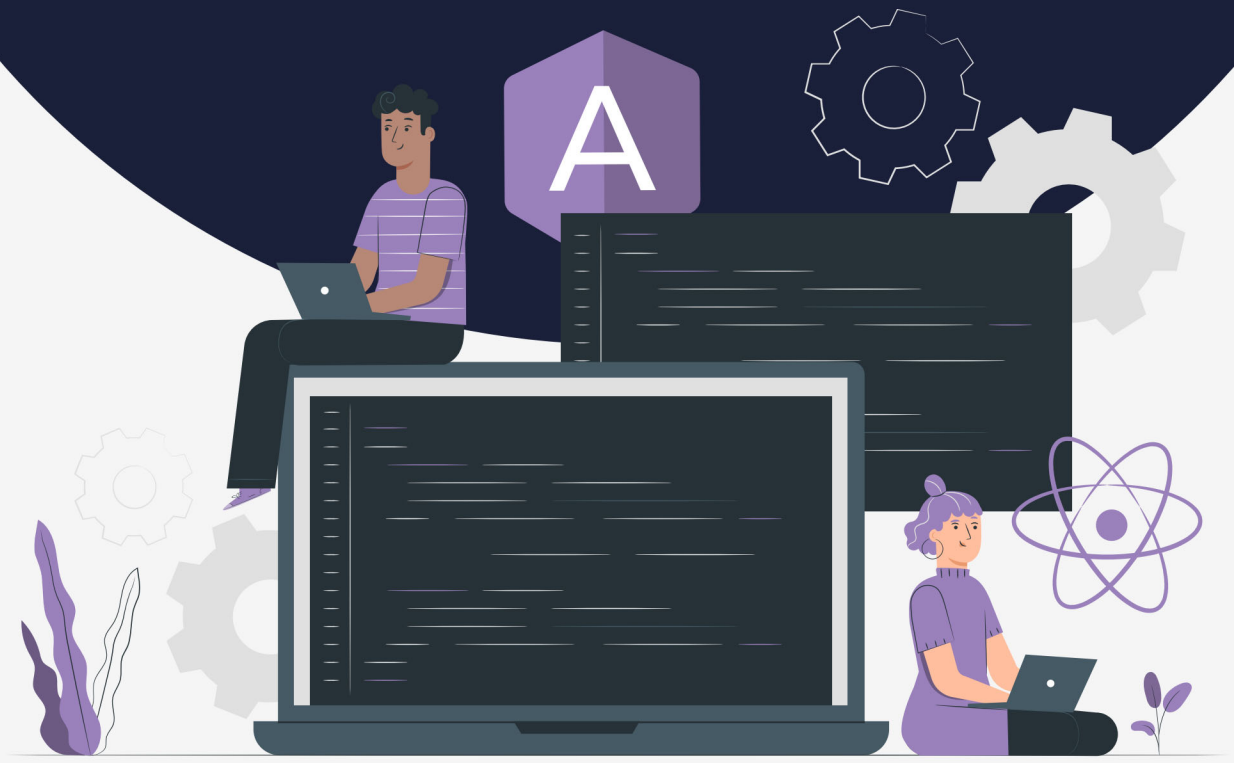
Lesson:

Number system



Pre-Requisites

- Basic Syntax of Java
- Variables, keywords, identifiers
- Conditionals
- Loops

List of Concepts Involved

- Number System - Decimal and Binary Numbers
- Conversion of Binary to Decimal
- Conversion of Decimal to Binary

Topic: Number System

Number system is used to represent mathematical numbers in different forms with different base values. Base value, as we all know is defined as a set of digits used to represent a number. Depending on the base value, number systems are of many types. Here, we are going to explore only two of the relevant ones, i.e., decimal and binary.
The decimal number system is the one having base 10 which means that any number can be represented in terms of 0 to 9 digits and powers of 10. All the numbers that we use in real life transactions are decimal numbers.

Decimal numbers Representation (Base 10) :

- $3451 = (3 \times 10^3) + (4 \times 10^2) + (5 \times 10^1) + (1 \times 10^0)$
- $576 = (5 \times 10^2) + (7 \times 10^1) + (6 \times 10^0)$

Here, as you can see, in the decimal representation the number is represented in powers of 10. It starts with the unit place digit having a value multiplied by the 0th power of 10 and as we move towards the left, the power of 10 keeps on increasing.

This same number can also be represented in terms of other number systems like hexadecimal where the number is represented with the help of 0 to 15 digits with the unit place being multiplied by 16^0 and power of 16 keeps on increasing as we move towards left.

For binary number system, any number can be represented in powers of 2 with 0s and 1s as shown below:

Binary numbers Representation (Base 2) :

- $1001 = (1 \times 2^3) + (0 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$
- $110 = (1 \times 22) + (1 \times 21) + (0 \times 20)$
- $0100 = (0 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (0 \times 2^0)$

Let us now look at some of the most used conversions in the number system.

Topic: Conversion of Binary to Decimal

To convert a binary number to a decimal number, we need to multiply each digit by the respective power of two as shown.

For example,

- $(1001)_2 = (1 \times 2^3) + (0 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) = (9)_{10}$
- $(110)_2 = (1 \times 2^2) + (1 \times 2^1) + (0 \times 2^0) = (6)_{10}$
- $(0100)_2 = (0 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (0 \times 2^0) = (4)_{10}$

Explanation :
We take the unit digit and multiply it by 2^0, and then for the subsequent digits, we keep on increasing the power of 2 as shown above.

Parity Digit: Parity Digit of a number n refers to the remainder it gives when divided by 2. If that reminder is 0, then the parity of the number is even otherwise odd.

Algorithm Steps:
Let's say n is the input binary number.

- **Step 1:** Find the parity digit of n. This is then multiplied by the respective power of two.
- **Step 2:** Increase the power of two and divide n by 10.
- **Step 3:** Keep on repeating the steps until the n is greater than zero.

Code:

```
public static void binary_to_decimal() {
int n;
Scanner sc= new Scanner(System.in);
n=sc.nextInt();
int ans = 0;
// here pw is power of two, initially, it will be 2⁰ = 1
int pw = 1; // 2⁰
while (n > 0) {
 int untis_digit = n % 10;
 ans += pw * untis_digit;

 n /= 10; // updating the units place
 pw *= 2; // increasing the power of two
}

 System.out.println(ans);
}
```

```
Input :
110
0100
1001
```

```
Output :
6
4
9
```

Topic: Conversion of Decimal to Binary

To convert the number from Decimal to Binary, we determine whether the parity of the unit place is 1 or 0 and then keep on dividing the number by 2.

Algorithm Steps:
Let's say n is the input decimal number.

- **Step 1:** Find the remainder/parity digit when n is divided by 2. This will be the unit place of the binary number.
- **Step 2:** Divide n by 2.
- **Step 3:** Keep on repeating the steps until the n is greater than zero.

Code:

```
public static void decimal_to_binary() {
int n;
 Scanner sc= new Scanner(System.in);
 n=sc.nextInt();
 int ans = 0; // our final decimal number
 int pw = 1; // 10^0 = 1
// here pw represents the power of 10 which will be needed to increment the place of decimal
number

 while (n > 0) {
  int parity = n % 2;
  ans += pw * parity;  // add it to our decimal number
  pw *= 10;
  n /= 2;
 }

 System.out.println(ans);
}

Input :
5
10
15

Output :
101
1010
1111
```

That is all about the number system (that would be relevant to us)

Let us meet in the next lesson!

Till then, keep learning and keep exploring !!

Upcoming Class Teasers

- OOPs concept