



Mohammed Sanih

Computer Science and Engineering

Immediate Joiner and I intend to be a part of an organization where i can constantly learn and develop my technical skills and make best use of it for the growth of the organization.

✉ mssanilh905@gmail.com

📍 MANGALORE, INDIA

🌐 github.com/Mohammed-Sanih

☎ 8147577963

🌐 linkedin.com/in/Mohammed Sanih

EDUCATION

Bachelor of Engineering(CSE)

Bearys Institute of Technology

08/2019 - 06/2023,

CGPA: 7.20

Courses

- Computer Science and Engineering

Pre University Education

K. Pandyarajah Ballal PU College

05/2017 - 03/2019,

Percentage : 78%

Courses

- Science(PCMB)

SSLC

Hazrath Seyyid Madani English Medium High School

05/2016 - 04/2017,

Percentage: 80.5%

INTERNSHIP

Web Design and Development

Blueline Computers

08/2022 - 09/2022,

Mangalore

Website development company in Mangalore efficiently delivers creative,interactive web solution and digital marketing services

SKILLS



PROJECTS

Fake Currency Detection Using Machine Learning Algorithm

- Final year Main Project

School Fee Management System

- DBMS Mini Project

Age Calculator

- MAD Mini Project

Rubik's Cube

- Computer Graphics Mini Project

Online Job Portal

- Internship Project

CERTIFICATES

RPA Developer Foundation

- Diploma of Completion

LANGUAGES

English

Full Professional Proficiency

Kannada

Professional Working Proficiency

Hindi

Professional Working Proficiency

Malayalam

Limited Working Proficiency

INTERESTS

Swimming Footballer Traveling

Declaration:

I do hereby declare that the above information is true to the best of my knowledge.

UNARY NOT OPERATOR (!)

In Java, the ! is a unary logical operator known as the logical NOT operator. It is used to invert the value of a boolean expression. When applied to a boolean value, the operator returns the opposite boolean value. If the expression is true, the operator returns false, and if the expression is false, the operator returns true.

Example:

```
boolean isSunny = false;

boolean isRainy = !isSunny;

System.out.println("Is it raining? " + isRainy);
```

In this example, we have a boolean variable isSunny initialized with the value false. The exclamation mark ! is then used to invert the value of isSunny and assign it to the variable isRainy. So, isRainy will have the value true. Finally, we print the result, which will output "Is it raining? true" indicating that it is raining because isRainy is true.

Code:

```
class Main {

    public static void main(String[] args) {

        boolean isTrue = true;

        boolean isFalse = false;

        System.out.println("!isTrue: " + !isTrue);

        System.out.println("!isFalse: " + !isFalse);

    }

}
```

Output:

!isTrue: false

!isFalse: true

BITWISE COMPLEMENT OPERATOR (~):

In Java, the bitwise complement operator is represented by the tilde (~) symbol. It is a unary operator that performs the bitwise complement operation on an integer value, which flips all the bits of the value, converting 0s to 1s and vice versa. The result is the bitwise complement of the original value.

Example :

```
int number = 42; // Binary: 00101010  
  
int complement = ~number; // Bitwise complement: 11010101  
  
System.out.println(complement); // Output: -43
```

Output:

In this example, the variable number is assigned the decimal value 42, which is equivalent to the binary value 00101010. When we apply the bitwise complement operator (~) to number, it flips all the bits, resulting in the binary value 11010101, which corresponds to the decimal value -43.

BIG INTEGER :

In Java, a '**BigInteger**' is a class that allows you to perform mathematical operations on integers of arbitrary length. It is particularly useful when dealing with numbers that are too large to be represented using primitive data types like '**int**' or '**long**'. The '**BigInteger**' class provides methods for performing various arithmetic operations such as addition, subtraction, multiplication, and division on these large integers.

Note: To use '**BigInteger**' in Java, you need to import the '**java.math**' package.

Example:

```
import java.math.BigInteger;  
  
class BigIntegerExample {  
    public static void main(String[] args) {  
        BigInteger num1 = new BigInteger("12345678901234567890");  
        BigInteger num2 = new BigInteger("98765432109876543210");  
        BigInteger sum = num1.add(num2);  
        System.out.println("Sum: " + sum);  
        BigInteger difference = num2.subtract(num1);  
        System.out.println("Difference: " + difference);  
        BigInteger product = num1.multiply(num2);  
        System.out.println("Product: " + product);  
    }  
}
```

```

        BigInteger quotient = num2.divide(num1);
        System.out.println("Quotient: " + quotient);
    }
}

```

Output:

Sum: 111111111011111111100

Difference: 86419753208541975320

Product: 121932631137021795200993369985299338000

Quotient: 7999999999

BigInteger a; //32 bytes

