RAGHAVENDRA KOWTAL

P:+918867003588|:github.com/Raghvkowtal|raghvkowtal.github.io/Personal-Portfolio/|inkedin.com/in/raghavendra-kowtal-583b1921a/

EDUCATION

SDM COLLEGE OF ENGINEERING AND TECHNOLOGY Dharwad, KA
Bachelor of Engineering Aug 2019 - June 2023
Major in Electronics and Communications;Cumulative GPA: 7.23/10.0;
Relevant Coursework: Web Development, Software Engineering; Operating Systems; Algorithms; Artificial Intelligence

KARNATAKA SCIENCE COLLEGE Dharwad, KA
Pre-University Education
2017 - 2019

WORK EXPERIENCE

Patil Electric Works Pvt Ltd
Armature Design Intern Aug 2022 –Sep2022

- Design of Armature according to the customer's requirements (Quality check, Assembly.)
- Manufacturer of armatures, dual speed motors & validate wiring harness in Hubballi.

VIRAL FISSION
Student Ambassador Jun 2021 – Sep 2021

- Promoted products digitally on Social-media platforms.
- Managed a team of 15-members, Supervision of product promotion on online platforms.

UNIVERSITY PROJECTS

REAL-TIME FOOD ORDERING SYSTEM Aug 2022 – Mar 2023

- Designed and implemented a food ordering application using HTML, CSS, JavaScript, Node JS, My SQL.
- Rendered 3 interfaces namely Customer, Admin, Restaurant sides.
- Enabled users to order food using Intranet, where 3000+ people can order at same time. LINK- https://tinyurl.com/2jx4tjsc

REACT.JS PORTFOLIO WEBSITE Jan 2023 – Apr 2023

- Created and rendered a portfolio design that can be used anyone.
- Technologies used: React.JS, HTML, CSS, JavaScript. LINK- https://tinyurl.com/RECTP

REACT.SJ ONLINE EDUCATIONAL WEBSITE Dec 2022 – Mar 2023

- Engineered and executed an educational website using React JS, JavaScript, HTML, CSS.
- Provided options to choose from 3 courses. LINK- https://tinyurl.com/EDCWB

GESTURE TO SPEECH CONVERSION USING ML Sep 2022 – Dec 2022

- Built a model using CNN Algorithm by providing 4000 Hand-gesture Data set with an Accuracy of 97%.
- Technologies used: ML, AI, Open-CV, Python, CNN, Tensorflow, Keras. LINK- https://tinyurl.com/HGSRE

FACE RECOGNITION USING MACHINE LEARNING Sep 2021 – Dec 2021

- Built a model using Haar Cascade Algorithm by providing 5000 face-images Data-set with an Accuracy of 95%.
- Trained the model using data-set of 5000 images.

ACTIVITIES

PUBLISHING
Published a paper entitled “Real-Time Intranet Based Food Ordering System” in World Journal of Advanced Research and Reviews. Paper link-“https://wjarr.com/content/development-novel-real-time-intranet-based-food-ordering-system”.

ADDITIONAL

Programming Languages: HTML, CSS, JavaScript, Python.
Frameworks: React JS, Material –UI, Bootstrap, Node.JS, Tailwind CSS.
Developer Tools: VS Code, PyCharm, IntelliJ.
Certifications & Training: Programming Concepts with ‘C’(ISCT-2019), MERN Stack Web Application Development (2023).
Email- raghavendrakowtal@gmail.com

NAME- RAGHAVENDRA KOWTAL

EMAIL-raghavendrakowtal@gmail.com

BATCH- A2 (2 – 4pm)

ASSIGNMENT TOPIC-
BITWISE OPERATORS IN JAVA

BITWISE OPERATORS IN JAVA

INTRODUCTION:

OPERATORS-

Operators in Java are the symbols used for performing specific operations in Java. Operators make tasks like addition, multiplication, etc. which look easy although the implementation of these tasks is quite complex.

TYPES OF OPERATORS IN JAVA

- Arithmetic Operators
- Unary Operators
- Assignment Operator
- Relational Operators
- Logical Operators
- Ternary Operator
- Bitwise Operators
- Shift Operators
- instance of operator

BITWISE OPERATORS

These operators are used to perform the manipulation of individual bits of a number. They can be used with any of the integer types. They are used when performing update and query operations of the Binary indexed trees.

- &, Bitwise AND operator: returns bit by bit AND of input values.
- |, Bitwise OR operator: returns bit by bit OR of input values.
- ^, Bitwise XOR operator: returns bit-by-bit XOR of input values.
- ~, Bitwise Complement Operator: This is a unary operator which returns the one’s complement representation of the input value, i.e., with all bits inverted.

JAVA PROGRAM

```
class GFG {
    public static void main(String[] args)
    {
        int d = 0b1010;
        int e = 0b1100;
        System.out.println("d & e: " + (d & e));
        System.out.println("d | e: " + (d | e));
        System.out.println("d ^ e: " + (d ^ e));
        System.out.println("~d: " + (~d));
        System.out.println("d << 2: " + (d << 2));
        System.out.println("e >> 1: " + (e >> 1));
        System.out.println("e >>> 1: " + (e >>> 1));
    }
}
```

TYPES OF SHIFT OPERATORS IN JAVA

- Signed Left Shift << The left shift operator moves all bits by a given number of bits to the left.
- Signed Right Shift >> The right shift operator moves all bits by a given number of bits to the right.

SIGNED LEFT SHIFT OPERATOR IN JAVA

Calculating the value of number<<2 if number=2. When the value of a number is shifted to the left two places, the leftmost two bits are lost. The number has a value of two. 0010 is the binary representation of the number 2.

JAVA PROGRAM

```
class GFG {
    public static void main(String[] args)
    {
        int number = 2;
        int Ans = number << 2;
        System.out.println(Ans);
    }
}
```

SIGNED RIGHT SHIFT OPERATOR IN JAVA

The Right Shift Operator moves the bits of a number in a given number of places to the right. The >> sign represents the right shift operator, which is understood as double greater than. When you type x>>n, you tell the computer to move the bits x to the right n places.

When we shift a number to the right, the least significant bits (rightmost) are deleted, and the sign bit is filled in the most considerable place (leftmost).

When the value of a number is shifted to the right two places, the rightmost two bits are lost. The number has a value of eight. 1000 is the binary representation of the number 8. The following is an example of how to perform the right shift.

JAVA PROGRAM

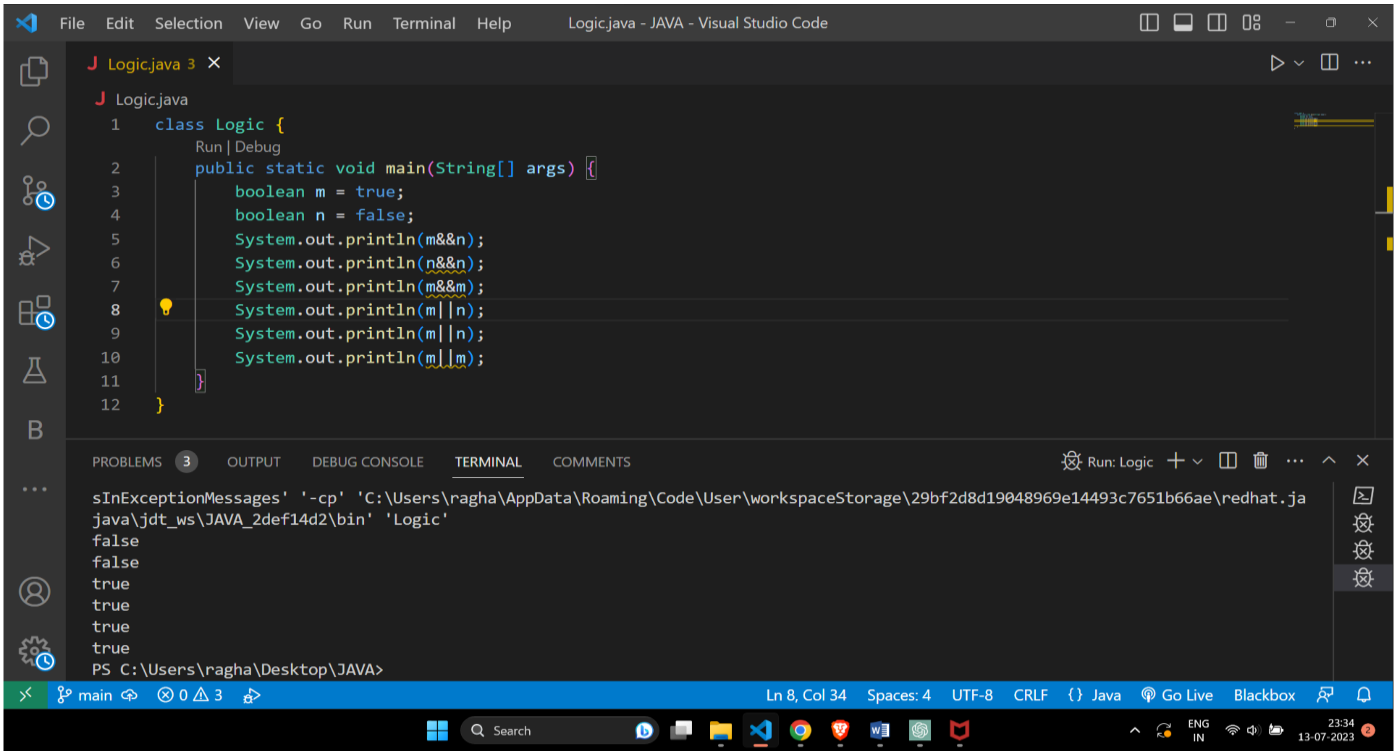
```
class GFG {
    public static void main (String[] args) {
       char hex[]={
       '0','1','2','3','4','5',
         '6','7','8','9','a','b','c',
         'd','e','f'
       };
      byte b=(byte) 0xf1;
      System.out.println("b = 0x" + hex [(b>>4) & 0x0f] + hex[b & 0x0f]);
    }
}
```

LOGICAL OPERATORS

```
class Logic {
    public static void main(String[] args) {
        boolean m = true;
        boolean n = false;
        System.out.println(m&&n);
        System.out.println(n&&n);
        System.out.println(m&&m);
        System.out.println(m||n);
        System.out.println(m||n);
        System.out.println(m||m);
    }
}
```

OUTPUT

```
false
false
true
true
true
true
```



```
class Logic {
    public static void main(String[] args) {
        boolean m = true;
        boolean n = false;
        System.out.println(m&&n);
        System.out.println(n&&n);
        System.out.println(m&&m);
        System.out.println(m||n);
        System.out.println(m||n);
        System.out.println(m||m);
    }
}
```

```
sInExceptionMessages' '-cp' 'C:\Users\ragha\AppData\Roaming\Code\User\workspaceStorage\29bf2d8d19048969e14493c7651b66ae\redhat.ja
java\jdt_ws\JAVA_2def14d2\bin' 'Logic'
false
false
true
true
true
true
PS C:\Users\ragha\Desktop\JAVA>
```