PALLAVI S

CONTACT

PHONE:
9353765475

EMAIL:
pallavi01suresh@gmail.com

LINKEDIN:
https://www.linkedin.com/in/pallavi-s-595906244

TECHNICAL SKILLS

- C
- Python
- Data structures in c
- Basics of DevOps

BEHAVIOURAL SKILL

- Ambivert
- Good listening skills
- Self Motivated
- Fast Learner
- Problem Solving
- Decision Making

LANGUAGES

- English
- Kannada
- Hindi

PROFILE

An undergraduate Computer Science and Engineer with keen interest in developing content. Eager to effectively utilize my skills and development of myself for the company in which I am working with utmost dedication.

EDUCATION

Bachelor's of Engineering , Adichunchangiri Institute of Technology,Chikkamagaluru
B.E. in Computer Science and Engineering 2019-2023
CGPA-8.87

PU ,Sri Bhuvanendra College,Karkala
Science PCMB 2017-2019
Percentage – 88.8

Govt.Junior College
10th Standard 2016-2017
Percentage-94.24

INTERNSHIPS/TRAINING

- Offline internship on **DevOps:** Some functions of AWS , Git , Github , EC2 , IAM , Apache Tomcat, Jenkins , Shell Scripting, Linux to Linux Connection. Project Title : Establishing of Virtual Private Cloud connection using AWS.
- An online internship on **Machine Learning with Python:** Accomplishment of project on the Sentiment Analysis on the dataset of the Restaurant Review using Multinomial Naive Bayes.

PROJECTS

- **Online Food Order:** Accomplishment of project on SQL and Web Development using HTML, CSS and JavaScript using Apache Server.
- **No Due Certificate app**: Accomplishment of project on Mobile application development using XML and Java on Android Studio.

ACHIEVEMENT/CERTIFICATION

- Workshop on Cyber Security and Blockchain Technology.
- Participated in a Hackathon program and presented a web development project on HTML and CSS.

DECLARATION

I hereby declare that the contents of my resume are accurate and I hold the responsibility of their authenticity and correctness.

Date: Place : Chikkamagaluru

BITWISE OPERATORS

Introduction to Bitwise Operators:

Bitwise operators are used in computer programming to manipulate individual bits of data at the binary level. They perform operations on the binary representation of integers or binary flags. Bitwise operators work on one or more binary digits (bits).

Some of the Bitwise operators are:

1. AND Operator (&):

The AND operator compares each bit of two integers and produces a new integer with a bit set if both corresponding bits in the operands are set. The result will have a bit set at a particular position only if both operands have bits set at that position. The truth table for the AND operator is as follows:

Operand 1	Operand 2	Result
0	0	0
0	1	0
1	0	0
1	1	1

Example:

int a = 5;   In binary, it can be represented as 0101

int b = 3;   In binary, it can be represented as 0011

int result = a & b;   In binary, it can be represented as 0001 (decimal: 1)

2. OR Operator (|):

The OR operator compares each bit of two integers and produces a new integer with a bit set if either of the corresponding bits in the operands is set. The result will have a bit set at a particular position if at least one of the operands has a bit set at that position. The truth table for the OR operator is as follows:

Operand 1	Operand 2	Result
0	0	0
0	1	1
1	0	1
1	1	1

Example:

int a = 5;   In binary, it can be represented as 0101

int b = 3;   In binary, it can be represented as 0011

int result = a | b;   In binary, it can be represented as 0111 (decimal: 7)

3. XOR Operator (^):

The XOR (exclusive OR) operator compares each bit of two integers and produces a new integer with a bit set if only one of the corresponding bits in the operands is set. The result will have a bit set at a particular position if the two operands have different bits at that position. The truth table for the XOR operator is as follows:

Operand 1	Operand 2	Result
0	0	0
0	1	1
1	0	1
1	1	0

Example:

int a = 5;   In binary, it can be represented as 0101

int b = 3;   In binary, it can be represented as 0011

int result = a ^ b;   In binary, it can be represented as 0110 (decimal: 6)

4. Left Shift Operator (<<):

The left shift operator shifts the bits of an integer to the left by a specified number of positions. It effectively multiplies the integer by 2 raised to the power of the number of positions shifted. The leftmost bits that are shifted out are lost, and zero bits are added on the right side. The syntax is as follows: `operand << number of positions`.

Example:

int a = 5;   In binary, it can be represented as 0101

int result= a<<1; In binary, it can be represented as 01010 (decimal: 10)

int result = a << 2;   In binary, it can be represented as 010100 (decimal: 20)

int result = a<<3; In binary, it can be represented as 0101000 (decimal: 40)

5. Right Shift Operator (>>):

The right shift operator shifts the bits of an integer to the right by a specified number of positions. It effectively divides the integer by 2 raised to the power of the number of positions shifted. The rightmost bits that are shifted out are lost. The syntax is as follows: `operand >> number of positions`

Example:

int a = 10;   In binary, it can be represented as 1010

int result = a >> 2;   In binary, it can be represented as 10 (decimal: 2)

int result = a>>3; In binary, it can be represented as 1 (decimal : 1)

6. Complement Operator (~):

The complement operator, also known as the NOT operator, flips each bit of an integer, converting 0 to 1 and 1 to 0. It is a unary operator and only operates on a single operand. The syntax is as follows: `~operand`.

Example:

int a = 5;   // binary: 0101

int result = ~a;   // binary: 1010 (decimal: -6)

In most programming languages, the complement operator produces the two's complement representation of negative numbers.