YANATI RAJANIVAS

CONTACT

rajanivas9573@gmail.com

+919573627556

www.linkedin.com/in/yanati-rajanivas-4ab36326a

1-164 Kalavakonda, Chillakur Mandal(%24412), Tirupati Diostrict.

SKILLS

- Python, Java Basics
- Html and Css
- Javascript
- Sql
- Database Fundamentals

EDUCATION

- **Bachelor Of Technology (6.8 CGPA)**

  NBKR Institute of Science and Technology
  2019 - 2023
  Vidyanagar,Nellore
  Computer Science Engineering

- **INTERMEDIATE (9.1 CGPA)**

  NBKR Science And Arts College
  2017 - 2019
  Vidyanagar,Nellore
  MPC

- **S.S.C (8.2 CGPA)**

  Z.P.P High School
  2016 - 2017
  Thikkavaram, Nellore

LANGUAGES

- English
- Telugu

CAREER OBJECTIVE

Highly motivated and enthusiastic to move in software industry. As a graduate seeking an opportunity to apply my knowledge, skills, and passion in a dynamic work environment. Eager to contribute to organization by utilizing my technical knowledge and gaining practical experience in work. Committed to continuous learning and growth, with a strong dedication to achieving professional excellence.

PROJECTS AND INTERNSHIPS

Projects

- **Design Of secured Authenticated Key Management Protocol For Cloud computing Environments**

  Front End : HTML , CSS, Javascript **Back End**: JSP

  Data base: MYSQL 5.0 **Server**: Apache Tomcat

Internships

- **Web Development using Html and Css**

  company : Think -Champ pvt.Ltd **Duration:** 1 Month

  Project : Flower Shop Through online

- **Completed internship in machine learning with python**

  company : Verzeo Edutech **Duration:** 1 Month

  Project: Heart Disease prediction using Machine Learning

ACHIEVEMENTS

- ✓ Second place winner in district level drawing competition
- ✓ Won second prize in ball batmenton game under district level

STRENGTHS

- ➢ Research and Analysis
- ➢ Quick learner
- ➢ Positivity and Adaptability
- ➢ Team Player

CERTIFICATIONS

- ❖ Python
- ❖ Html And Css
- ❖ Machine Learning
- ❖ Azure AI Fundamentals
- ❖ Database Administrator Fundamentals

Java Assignment7

1) Differences Between Else-if and Switch:

Both `else if` and `switch` are control structures in programming that allow you to make decisions based on different conditions. However, they have different use cases and syntax.

1. **else if**:
    - `else if` is used within an `if` statement to handle multiple conditions sequentially.
    - It's commonly used when you have a series of conditions that you want to check one after another, and each condition may or may not be related to the others.
    - It's more flexible in terms of complex conditions and ranges, as you can use logical operators (e.g., `&&`, `||`) and comparison operators (e.g., `<`, `>`, `==`) to build complex conditions.
    - Example in a programming language like Java:

Example Program:

```
package programs;

import java.util.Scanner;

public class ifelse {

	public static void main(String[] args) {
		Scanner scan = new Scanner(System.in);
		System.err.println("Enter the number1");
		int num1 = scan.nextInt();
		System.err.println("Enter the number2");
		int num2 = scan.nextInt();
		System.err.println("Enter the number3");
		int num3 = scan.nextInt();

		if (num1>num2 && num1>num3) {
			System.out.println("num1 is greater");
		}
		else if(num2>num3 && num2>num1){
			System.out.println("num2 is greater");
		}
		else {
			System.out.println("num3 is greater");
		}

	}

}
```

Output:

```
Enter the number1
25
Enter the number2
30
Enter the number3
60
num3 is greater
```

switch:

- `switch` is used when you have a single expression whose value is compared to multiple possible cases.
- It's useful when you're comparing a single value against multiple constant values, and each case typically performs a specific action.
- It's often cleaner and more readable when you have a simple comparison with constant values.
- Each case must end with a `break` statement to exit the switch block.
- Example in a programming language like C++:

Example program:

```
package programs;

import java.util.Scanner;

public class Switch {

	public static void main(String[] args) {
		Scanner scan = new Scanner(System.in);
		System.out.println("Enter the number");
		int num  = scan.nextInt();

		switch(num) {

		case 1:
			System.out.println("Sunday");
			break;

		case 2:
			System.out.println("Monday");
			break;
		case 3:
			System.out.println("Tuesday");
			break;
		case 4:
			System.out.println("Wednesday");
```

```
                break;
            case 5:
                System.out.println("thursday");
                break;
            case 6:
                System.out.println("Friday");
                break;
            case 7:
                System.out.println("saturday");
                break;
            }

        }

}

Output:
Enter the number
6
Friday
```

2)Nested if and Nested if-else:

I. Nested If:

Certainly! Nested `if` statements in Java allow you to have an `if` statement inside another `if` statement. This can be useful when you need to evaluate multiple conditions in a hierarchical manner.

Here's a more detailed explanation of nested `if` statements with an example in Java:

Suppose you're building a program that helps a user determine their eligibility for a scholarship based on their academic performance and financial need. You want to provide different messages based on these factors.

Example Program:

```
package programs;

public class Nestedif {

public static void main(String[] args) {
        int temperature = 31;
        boolean isRainy = false;

        if (temperature > 30) {
            System.out.println("It's hot.Wear light clothes.");
        if (isRainy) {
            System.out.println("Don't forget to carry an umbrella.");
        }
    }
```

```
	}

}
```

Output:

```
It's hot outside. Wear light and breathable clothes.
```

Nested if-else:

Certainly! Nested `if` statements in Java allow you to place one `if` statement inside another, creating a hierarchy of conditional checks. This allows you to evaluate multiple conditions in a structured manner. Let's go through the concept with an example:

Suppose you are writing a program that helps a user determine what type of vacation they should take based on their preferences and budget.

Example Program:

```
package programs;

public class Nestedifelse {

	public static void main(String[] args) {
		boolean hasPassport = true;
	  double budget = 1500;

	  if (hasPassport) {
		System.out.println("You have a passport.");

		if (budget >= 2000) {
		    System.out.println("consider an international vacation.");
		}
		else {
		System.out.println("You don't have a passport.");
		System.out.println("need to apply for a passport first.");
		}
	  }
	  else {
		System.out.println("Nothing");
	  }

	}

}
```

Output:

```
You have a passport.
You don't have a passport.
You might need to apply for a passport first.
```