

Below is a simple Python program that calculates a person's zodiac sign and gives a brief interpretation according to Indian astrology (Vedic astrology). Indian astrology primarily relies on the Moon sign and the position of planets at the time of birth. For the sake of simplicity, this example will use the Sun sign basis. To implement this, we first need to know the person's date of birth, and then we can ascertain their Sun sign according to Vedic astrology. Here's a simple Python program to compute the Sun sign based on the birth date:

```
In [1]: from datetime import datetime

def get_zodiac_sign(date_of_birth):
    # Zodiac signs and their date ranges
    zodiac_signs = [
        ("Capricorn", (datetime(1, 1, 1), datetime(1, 1, 19))), # January 1 - January 19
        ("Aquarius", (datetime(1, 1, 20), datetime(1, 2, 18))), # January 20 - February 18
        ("Pisces", (datetime(1, 2, 19), datetime(1, 3, 20))), # February 19 - March 20
        ("Aries", (datetime(1, 3, 21), datetime(1, 4, 19))), # March 21 - April 19
        ("Taurus", (datetime(1, 4, 20), datetime(1, 5, 20))), # April 20 - May 20
        ("Gemini", (datetime(1, 5, 21), datetime(1, 6, 20))), # May 21 - June 20
        ("Cancer", (datetime(1, 6, 21), datetime(1, 7, 22))), # June 21 - July 22
        ("Leo", (datetime(1, 7, 23), datetime(1, 8, 22))), # July 23 - August 22
        ("Virgo", (datetime(1, 8, 23), datetime(1, 9, 22))), # August 23 - September 22
        ("Libra", (datetime(1, 9, 23), datetime(1, 10, 22))), # September 23 - October 22
        ("Scorpio", (datetime(1, 10, 23), datetime(1, 11, 21))), # October 23 - November 21
        ("Sagittarius", (datetime(1, 11, 22), datetime(1, 12, 21))), # November 22 - December 21
        ("Capricorn", (datetime(1, 12, 22), datetime(1, 12, 31))) # December 22 - December 31
    ]

    # Extract the month and day for comparison
    birth_date = datetime.strptime(date_of_birth, '%Y-%m-%d')
    month_day = datetime(1, birth_date.month, birth_date.day)

    # Determine zodiac sign
    for sign, (start_date, end_date) in zodiac_signs:
        if start_date.month == end_date.month: # Single month zodiac
            if month_day.month == start_date.month and start_date.day <= month_day.day <= end_date.day:
                return sign
        else: # Cross month zodiac (Capricorn and Sagittarius)
            if (month_day.month == start_date.month and month_day.day >= start_date.day) or \
                (month_day.month == end_date.month and month_day.day <= end_date.day):
                return sign

    return None
```

```
def main():
    # User input for date of birth
    dob = input("Enter your date of birth (YYYY-MM-DD): ")
    zodiac_sign = get_zodiac_sign(dob)

    if zodiac_sign:
        print(f"Your zodiac sign is: {zodiac_sign}")
    else:
        print("Invalid date of birth entered.")

if __name__ == "__main__":
    main()
```

Enter your date of birth (YYYY-MM-DD): 1983-05-13

Your zodiac sign is: Taurus

The function `get_zodiac_sign` retrieves the Zodiac sign from the list based on the birth date provided. Make sure that you handle potential errors (like incorrect date formats) in a more robust implementation. Note This simple program only calculates the Sun sign. For a more accurate calculation according to Indian astrology (which takes into account the Moon sign and the exact position of the planets), you would typically use a library like `AstroPy` or generate a Kundli using comprehensive astrological calculations.

Vedic astrology prediction tool in Python requires knowledge about astrological principles, astronomical computations, and the respective planetary positions for a specific date, time, and location. For a basic implementation, we can create a program that uses the birth date, time, and location to make predictions related to a person's personality traits, career, relationships, etc., generally based on their Moon sign or Lagna (Ascendant) based on Vedic astrology.

While creating a full-fledged Vedic astrology prediction tool is quite complex and requires extensive datasets on planetary positions, we can create a very simplified version instead.

Here's a Python program that takes a person's birth details (date, time, and location) to determine their Moon sign and make a basic personality prediction based on that sign.

Note: For a real-world application, you would ideally use a library or API like `pyswisseph` (a Python wrapper for the Swiss Ephemeris) to calculate planetary positions accurately.

Here's a simplified version of a horoscope calculation tool using Moon signs only:

```
In [2]: from datetime import datetime

        # Dictionary to define Moon sign predictions
```

```

moon_sign_predictions = {
    "Aries": "You are energetic, adventurous, and pioneering. You thrive in competitive environments.",
    "Taurus": "You are practical, stable, and reliable. You cherish life's pleasures and comforts.",
    "Gemini": "You are adaptable, communicative, and curious. A social butterfly, you enjoy learning.",
    "Cancer": "You are intuitive, emotional, and nurturing. Family and home hold great importance for you.",
    "Leo": "You are confident, ambitious, and charismatic. You thrive on being in the spotlight.",
    "Virgo": "You are analytical, meticulous, and service-oriented. You excel at problem-solving.",
    "Libra": "You are diplomatic, sociable, and peace-loving. You thrive in harmonious environments.",
    "Scorpio": "You are passionate, resourceful, and intense. You hold deep emotions and are highly perceptive.",
    "Sagittarius": "You are optimistic, freedom-loving, and explorative. Travel and adventure excite you.",
    "Capricorn": "You are disciplined, responsible, and ambitious. You strive for achievement and success.",
    "Aquarius": "You are innovative, unique, and humanitarian. You embrace change and the unconventional.",
    "Pisces": "You are compassionate, artistic, and sensitive. You are often in touch with the spiritual side."
}

def calculate_moon_sign(dob):
    """ A simplified function that returns the Moon sign based on the birth date. """
    # For demonstration, we use only birth month and day to determine Moon sign directly.
    # A real calculation would need actual Moon position data.
    dob_month_day = datetime.strptime(dob, '%Y-%m-%d')
    month_day = (dob_month_day.month, dob_month_day.day)

    # Determine Moon sign based on simplified data (for illustration only)
    if (3, 21) <= month_day <= (4, 19):
        return "Aries"
    elif (4, 20) <= month_day <= (5, 20):
        return "Taurus"
    elif (5, 21) <= month_day <= (6, 20):
        return "Gemini"
    elif (6, 21) <= month_day <= (7, 22):
        return "Cancer"
    elif (7, 23) <= month_day <= (8, 22):
        return "Leo"
    elif (8, 23) <= month_day <= (9, 22):
        return "Virgo"
    elif (9, 23) <= month_day <= (10, 22):
        return "Libra"
    elif (10, 23) <= month_day <= (11, 21):
        return "Scorpio"
    elif (11, 22) <= month_day <= (12, 21):
        return "Sagittarius"
    elif (12, 22) <= month_day <= (1, 19):
        return "Capricorn"
    elif (1, 20) <= month_day <= (2, 18):

```

```
        return "Aquarius"
    elif (2, 19) <= month_day <= (3, 20):
        return "Pisces"

    return None

def main():
    # User input for date of birth
    dob = input("Enter your date of birth (YYYY-MM-DD): ")

    # Calculate Moon sign
    moon_sign = calculate_moon_sign(dob)

    if moon_sign:
        print(f"Your Moon sign is: {moon_sign}")
        print("Prediction for you:")
        print(moon_sign_predictions[moon_sign])
    else:
        print("Invalid date of birth entered.")

if __name__ == "__main__":
    main()
```

Enter your date of birth (YYYY-MM-DD): 1983-05-13

Your Moon sign is: Taurus

Prediction for you:

You are practical, stable, and reliable. You cherish life's pleasures and comforts.

In []: