Programming with Python (3) Jan 21, 2025

Brief Summary of contents discussed.

Control Structures

Programs for real life problems often require non-sequential and repetitive execution of instructions. Python provides control structures to achieve non-sequential execution and repetitive executions using constructs such as ‘if-else’, ‘for’, and ‘while’

if Conditional Statement: used for selecting next set of instructions to be executed based on condition(s)

Some common forms of ‘if’ Conditional

```
1.  if < condition >:
            < Sequence of statements to be executed >
```

Note: *notice the placement of ‘:’* and the *space (indentation)* before the statements to be executed if condition is True.

```
2.  if <condition >:
            < Sequence of statements to be executed >
    else:
            < Sequence of statements to be executed >


3.  if <condition 1 > :
            < Sequence of statements to be executed >
    elif <condition 2> :
                    < Sequence of statements to be executed >
    elif  <condition 3> :
            < Sequence of statements to be executed >
    .
    .
    else:
            < Sequence of statements to be executed >
```

4. Nested if-elif-else

```
if <condition 1 >:
    if <condition 2 >:
        < Sequence of statements to be executed >
    else:
        < Sequence of statements to be executed >
    < Sequence of statements to be executed >
elif <condition>:
    < Sequence of statements to be executed >
else:
    < Sequence of statements to be executed >
```

There can be different combinations in which if-else is used. Try to get comfortable using (2) and (3) above first. (Note: Flat is better than Nested -> The Zen of Python)

Programming Exercise (Use only constructs discussed in the class so far):

1. **Odd or Even**: Check if a number is even or odd.
2. **Grade Calculator**: Write a program to input marks from the student and assign grade to a student on the basis of marks obtained
    i. [85-100] Grade: ‘A’
    ii. [70 - 84] Grade: ‘B’
    iii. [50 - 69] Grade: ‘C’
    iv. [0 - 49] Grade: ‘D’
3. **Moderate Marks**: Input marks obtained and moderate marks by 1 or 2 marks to achieve passing marks. (Take suitable value of pass marks).
4. **Max of 3 numbers**: Input 3 numbers n1, n2 and n3. Find the max of them.
5. **Positive, Negative or Zero**: Input a number from the user and check whether the number is positive, negative or zero.
6. **Leap Year**: Input a year from the user. Using control structures, identify if the year is leap or not. A year is a leap year if it is divisible by 4 **and** (not divisible by 100 **or** divisible by 400).
    a. Try writing the different conditions as separate conditions.
    b. Try writing all conditions as a single condition.

7. **Electricity Bill Generation:** Write a program to calculate the electricity bill based on the following rules:
    a. Units ≤ 100: ₹5 per unit.
    b. 101–300 units: ₹5 for the first 100 units + ₹7 for the next 200 units.
    c. Units > 300: ₹5 for the first 100 + ₹7 for the next 200 + ₹10 for units above 300.
8. **Triangle Type:** Write a program that takes three sides of a triangle as input and determines if it forms:
    a. An Equilateral triangle (all sides equal),
    b. An Isosceles triangle (two sides equal), or
    c. A Scalene triangle (all sides different). Additionally, check if the input forms a valid triangle (sum of any two sides should be greater than the third side).