

CONDITIONAL STATEMENTS.

- Used when decision making is required.
- "AGR" 50,000 hai ya us say zada tou discount
- IF Bill \geq 50,000 THEN

Discount \leftarrow 50%.

ELSE

pay price

Question: Input a number from the User and check if it is equal to a predefined.

DECLARE Num AS INTEGER
DECLARE predefinedVal AS INTEGER

INPUT Num

predefined ← 50

IF Num = predefined THEN
 OUTPUT "Both values are equal"
ELSE
 OUTPUT "Both values are not
 equal"

Question 2 Input 2 Numbers from User and Output "Equal" if they are equal and outputs "Not Equal" If they are not equal.

DECLARE Num1: INT
DECLARE Num2: INT

INPUT Num1

INPUT Num2

IF Num1 = Num2 THEN
 OUTPUT "Equal"
ELSE
 OUTPUT "Not Equal"

Revision

Conditional Statements

- Variables: A data structure to store data.
- How to make a variable
 - **DECLARE** **VariableName**
 - e.g. **DECLARE** **count**

$\text{sum} = \text{num}_1 > \text{not}$
 $\text{sum} \leftarrow \text{num}_1 < \text{same}$

Question Write a program that inputs 3 numbers and output their sum.

```
→  
    DECLARE Num1: INTEGER  
    DECLARE Num2: INTEGER  
    DECLARE sum: INTEGER  
    OUTPUT "Enter a number"  
    INPUT Num1  
    OUTPUT "Enter second number"  
    INPUT Num2  
    sum  $\leftarrow$  num1 + num2  
    OUTPUT sum.
```

Conditional Statements

Question Prompt a student to enter His/Her marks. If the marks are below 45 then output "FAIL" if not then output "PASS"

```
DECLARE Marks()REAL  
OUTPUT "Enter Marks"  
INPUT Marks  
IF Marks < 45 THEN()  
    OUTPUT " FAIL"  
ELSE  
    OUTPUT " PASS"
```

Question 2 Prompt the user to enter a number
Output Positive if it is positive or if it is negative then print negative.

```
DECLARE Num: INTEGER  
OUTPUT "ENTER Number"  
INPUT Num  
IF Num <= 0 THEN:  
    OUTPUT "Negative"  
ELSE  
    OUTPUT "Positive"
```