

**Algorithm to Check if a Year is a Leap Year**

1. **Start**
2. **Prompt** the user to enter a year.
3. **Read** the input and convert it to an integer, store it in the variable year.
4. **Check** if the year is divisible by 4 **and** not divisible by 100:
   * If **true**, go to step 6.
   * If **false**, go to step 5.
5. **Check** if the year is divisible by 400:
   * If **true**, go to step 6.
   * If **false**, go to step 7.
6. **Print** "year is a leap year."
7. **Print** "year is not a leap year."
8. **End**

**YES**

if year divisible by 4 & not divisible by 100

**YES**

**NO**

**Output:** "year is not a leap year."

if year divisible by 400

**NO**

Output**:** "year is a leap year."

**Store** the input year in integer Variables. .Celsius

Input**:** *enter year*

Program 2: Write a program to check if the year entered by the user is a leap year or not.

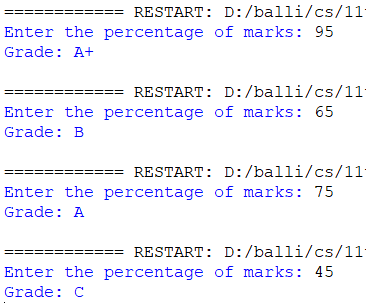
Chapter-4 Control Statements

**Unit -II Python programming**

|  |  |
| --- | --- |
|  | Program 2: Write a program to check if the year entered by the user is a leap year or not.  Date  Page No. |
|  |  |
|  | def main():  # Get input from the user  year = int(input("Enter a year: "))  # Check if it is a leap year  if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):  print( f"{year} is a leap year." )  else:  print( f"{year} is not a leap year." )  # Call the main function  if \_\_name\_\_ == "\_\_main\_\_":  main() |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Chapter-4 Control Statements

**Unit -II Python programming**



Program 3: Write a program to input the percentage of marks of a student and then find the grade of student as per the conditions given below.

Percentage of Marks Grade

Above 90 A+

70 to 90 A

50 to 70 B

Below 50 C

Output**: Grade = "A+"**

**Store** the input percentage in Variables. .Celsius

if

percentage >90

Input**:** *enter Percentage*

Output**: Grade = "A"**

if

percentage >50

if

percentage >70

Output**: Grade = "B"**

Output**: Grade = "C"**

**NO**

**NO**

**NO**

**YES**

**YES**

**YES**

**Algorithm to Determine Student Grade Based on Percentage**

1. **Start**
2. **Prompt** the user to enter the percentage of marks.
3. **Read** and store the percentage in a variable percentage.
4. **Check** if percentage > 90
   * If true, assign grade = "A+" and go to step 8.
5. **Check** if percentage >= 70
   * If true, assign grade = "A" and go to step 8.
6. **Check** if percentage >= 50
   * If true, assign grade = "B" and go to step 8.
7. If none of the above, assign grade = "C"
8. **Display** the grade.
9. **End**

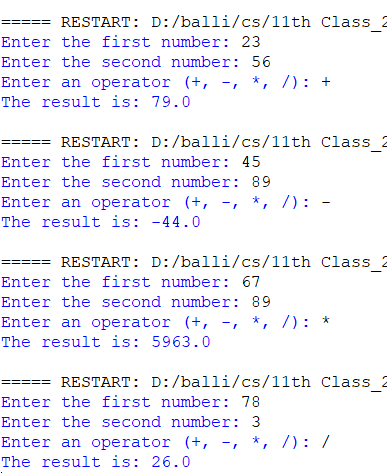
Chapter-4 Control Statements

**Unit -II Python programming**

if

percentage >90

|  |  |
| --- | --- |
|  | Program 3: Write a program to input the percentage of marks of a student and then find the grade of student as per the conditions given below.  Percentage of Marks Grade  Above 90 A+  70 to 90 A  50 to 70 B  Below 50 C  Chapter-4 Control Statements  **Unit -II Python programming**  Date  Page No. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | def main():  percentage = float(input( "Enter the percentage of marks: " ))  # Determine the grade  if percentage > 90:  grade = "A+"  elif percentage >= 70:  grade = "A"  elif percentage >= 50:  grade = "B"  else:  grade = "C"  # Display the result  print( f"Grade: {grade}" )  # Call the main function  if \_\_name\_\_ == "\_\_main\_\_":  main() |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |



**Algorithm: Simple Calculator**

1. **Start**
2. **Prompt** the user to enter the first number.
3. **Store** the input as num1 after converting it to a float.
4. **Prompt** the user to enter the second number.
5. **Store** the input as num2 after converting it to a float.
6. **Prompt** the user to enter an operator (+, -, \*, or /).
7. **Store** the input as operator.
8. **Check** the value of operator:
   * If operator is '+', then:
     + Calculate result = num1 + num2
     + **Display** the result.
   * Else if operator is '-', then:
     + Calculate result = num1 - num2
     + **Display** the result.
   * Else if operator is '\*', then:
     + Calculate result = num1 \* num2
     + **Display** the result.
   * Else if operator is '/', then:
     + - * calculate result = num1 / num2
         * **display** the result.
9. **End**

print num1/num2

print num1\*num2

print num1-num2

print num1+num2

**YES**

**YES**

**YES**

**YES**

**NO**

**NO**

**NO**

**NO**

if operator is “/”

if operator is “\*”

if operator is “\*”

if operator is “\*”

if operator is “-”

if operator is “+”

***Store*** *the operator in operator variable*

Input**:** *enter an operator (+,-,\*,/)*

***Store*** *the first and second number in variable num1 and num2*

Input**:** *enter the first and Second number*

***Store*** *the first and second number in variable num1 and num2*

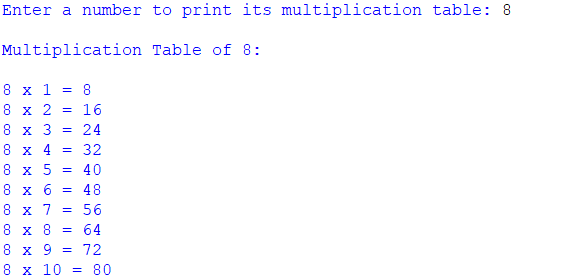
Input**:** *enter the first and Second number*

Chapter-4 Control Statements

Program 4: Write a program to create a simple calculator to perform basic arithmetic operations on two numbers. The program should do the following:

1. Accept two numbers from the user.
2. Ask user to input any of the operator(+,-,\*,/). An error message is displayed if the user enters anything else.
3. As per the input of operator, do the corresponding operation on numbers entered by user.

|  |  |
| --- | --- |
| Program 4: Write a program to create a simple calculator to perform basic arithmetic operations on two numbers. The program should do the following:   1. Accept two numbers from the user. 2. Ask user to input any of the operator(+,-,\*,/). An error message is displayed if the user enters anything else. 3. As per the input of operator, do the corresponding operation on numbers entered by user. | Chapter-4 Control Statements  Date  Page No. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | def main():  num1 = float(input( "Enter the first number: " ))  num2 = float(input( "Enter the second number: " ))  # Ask for an operator  operator = input( "Enter an operator (+, -, \*, /): " )  # Perform the corresponding operation  if operator == '+':  result = num1 + num2  print( f"The result is: {result}" )  elif operator == '-':  result = num1 - num2  print( f"The result is: {result}" )  elif operator == '\*':  result = num1 \* num2  print( f"The result is: {result}" )  elif operator == '/':  result = num1 / num2  print( f"The result is: {result}" )  if \_\_name\_\_ == "\_\_main\_\_":  main() |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |



*i >10*

**Display result in format: number x i = result**

**Calculate result = number \* i**

For i = 1 to 10, repeat:

**Display heading "Multiplication Table of number**

***Read the number and store in variable 'number'***

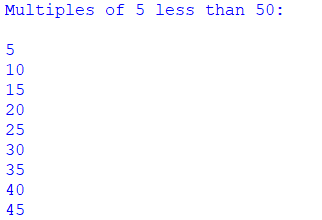
Input**: *Prompt user to enter a number***

**Algorithm: Print Multiplication Table of a Given Number**

1. **Start**
2. **Prompt** the user to enter a number.
3. **Read** the number and store it in a variable number.
4. **Display** a heading: "Multiplication Table of number".
5. **Repeat** the following steps from i = 1 to i = 10:
   * Calculate result = number \* i
   * Display the result in the format: number x i = result
6. **End**

Program 1: Write a program to print the table of a given number. The number has to be entered by the user.

|  |  |
| --- | --- |
|  | Program 1: Write a program to print the table of a given number. The number has to be entered by the user.  Chapter-4 Control Statements  Date  Page No. |
|  |  |
|  | def main():  # Ask the user to enter a number  number = int(input("Enter a number to print its multiplication table: "))  print(f"\nMultiplication Table of {number}:\n")  # Print the multiplication table from 1 to 10  for i in range(1, 11):  print(f"{number} x {i} = {number \* i}")  # Call the main function  if \_\_name\_\_ == "\_\_main\_\_":  main() |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |



*Yes*

*No*

1. *If true, print the value of i*
2. *Increment i by 5*
3. *Loop back to the condition i < 50*

Check if

i < 50

*initialize a loop with variable i = 5*

**

Display the message: "Multiples of 5 less than 50:"

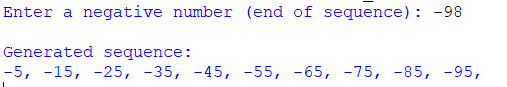
**Algorithm: Print Multiples of 5 Less Than 50**

1. **Start**
2. **Display** the message: "Multiples of 5 less than 50:"
3. **Set** the loop to start at 5 and go up to (but not including) 50, incrementing by 5
4. **Repeat** the following steps for each value i in the range 5 to 45:
   * **Print** the value of i
5. **End**

Chapter-4 Control Statements

Program 2: Write a program to print all multiple of 5 that are smaller than 50 with the help of range function with for loop

|  |  |
| --- | --- |
|  | Program 2: Write a program to print all multiple of 5 that are smaller than 50 with the help of range function with for loop  Chapter-4 Control Statements  Date  Page No. |
|  |  |
|  | def main():  print("Multiples of 5 less than 50:\n")  # Use range with a step of 5, starting from 5 up to (but not including) 50  for i in range(5, 50, 5):  print(i)  # Call the main function  if \_\_name\_\_ == "\_\_main\_\_":  main() |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |



*No*

*No*

*Yes*

*Yes*

***→ Process: i = i - 10  
→ back to decision block.***

***Output: Print i***

Check if

i ≥ n?

***Process:*** *i = -5*

Display 'Generated sequence:

Check if

n<0

*Store input in variable n.*

**

Enter a negative number (n)

**Algorithm: Generate Sequence -5, -15, -25, ..., up to n**

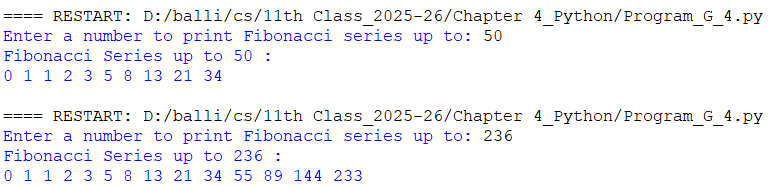
1. **Start**
2. **Prompt** the user to enter a negative number n.
3. **Read** the input and store it in variable n.
4. **If** n is not negative, **display** an error message and **end** the program.
5. **Display** the message: "Generated sequence:"
6. **Initialize** a loop starting from -5, decreasing by 10, until the value is less than or equal to n:
   * **Print** the current value.
7. **End**

Program 3: Write a program to generate the sequence :

-5,-15,-25….. upto n

Where n is a negative number input by the user.

|  |  |
| --- | --- |
|  | Program 3: Write a program to generate the sequence :  -5,-15,-25….. upto n  Where n is a negative number input by the user.      Chapter-4 Control Statements  Date  Page No. |
|  |  |
|  |  |
|  | def main():  # Ask the user to enter a negative number  n = int(input("Enter a negative number (end of sequence): "))  # Check if input is negative  if n >= 0:  print("Please enter a negative number.")  return  print("\nGenerated sequence:")  # Start from -5 and decrement by 10 until reaching or passing n  for i in range(-5, n - 1, -10):  print(i, end=", ")  # Call the main function  if \_\_name\_\_ == "\_\_main\_\_":  main() |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |



*Yes*

*No*

Print a newline

*Calculate temp = a + b*

*Set a = b , Set b = temp*

Print the value of a

(on the same line)

While a ≤ limit

Display the message:

“Fibonacci Series up to", limit

*Initialize variables: a = 0, b = 1*

*store in variable limit*

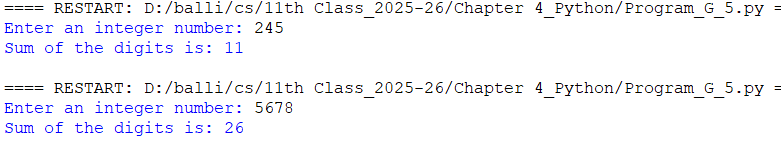
Prompt user to enter a number

**Algorithm: Print Fibonacci Series up to a Given Number**

1. **Start**
2. Prompt the user to **enter a number** and **store it in limit**
3. Initialize two variables:  
   → a ← 0  
   → b ← 1
4. Display the message: "Fibonacci Series up to", limit, ":"
5. **Repeat while a is less than or equal to limit:**
   1. Print the value of a (on the same line)
   2. Compute the next Fibonacci number:  
      → temp ← a + b
   3. Update values:  
      → a ← b  
      → b ← temp
6. Print a newline (for formatting)
7. **End**

Program 4: Write a program to print the Fibonacci Series up to a number entered by the user.

|  |  |
| --- | --- |
|  | Program 4: Write a program to print the Fibonacci Series up to a number entered by the user.  Chapter-4 Control Statements  Date  Page No. |
|  |  |
|  | def main():  # Program to print the Fibonacci series up to a given number  # Get the upper limit from the user  limit = int(input("Enter a number to print Fibonacci series up to: "))  # Initialize the first two Fibonacci numbers  a, b = 0, 1  print("Fibonacci Series up to", limit, ":")  # Print Fibonacci numbers while the next number is less than or equal to the limit  while a <= limit:  print(a, end=' ')  a, b = b, a + b # Update values  print() # Newline after the series  # Call the main function  if \_\_name\_\_ == "\_\_main\_\_":  main() |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |



*Yes*

*No*

**Output "Sum of digits is: digit\_sum"**

***digit ← num % 10***

***digit\_sum ← digit\_sum + digit***

***num ← num // 10***

*** Go back to check num > 0***

Check if ,is num > 0?

***num ← |num|***

*(take absolute value)*

***digit\_sum ← 0***

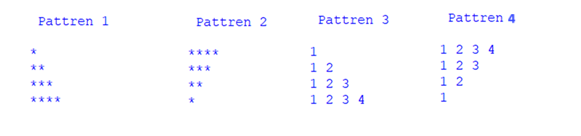
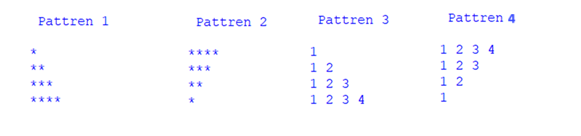
**Input num** (prompt user to enter an integer)

**Algorithm: Sum of Digits of an Integer**

1. **Start**
2. Prompt the user to **enter an integer number** and store it in num
3. Take the **absolute value** of num to handle negative input (i.e., num ← |num|)
4. Initialize digit\_sum ← 0
5. **Repeat while** num > 0:
   1. digit ← num % 10                 # Extract the last digit
   2. digit\_sum ← digit\_sum + digit      # Add digit to the sum
   3. num ← num // 10                   # Remove the last digit
6. Print "Sum of the digits is: ", digit\_sum
7. **End**

Program 5: Write a program to print the sum of digit of an integer number, input by the user.

|  |  |
| --- | --- |
|  | Program 5: Write a program to print the sum of digit of an integer number , input by the user.  Chapter-4 Control Statements  Date  Page No. |
|  |  |
|  | def main():  # Program to calculate the sum of digits of a number  # Get the number from the user  num = int(input("Enter an integer number: "))  # Make sure we handle negative numbers too  num = abs(num)  # Initialize sum variable  digit\_sum = 0  # Loop to extract and sum the digits  while num > 0:  digit = num % 10 # Get the last digit  digit\_sum += digit # Add digit to the sum  num = num // 10 # Remove the last digit  # Print the result  print("Sum of the digits is:", digit\_sum)  # Call the main function  if \_\_name\_\_ == "\_\_main\_\_":  main() |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |



Program 6: Write a program to print Patterns.

*True*

**Algorithm for the Pattern Printing Program**

**1. Start**

**2. Set rows ← 4**

**Pattern 1: Increasing Asterisks**

1. Print a heading: "Pattern 1"
2. Repeat for i from 1 to rows (inclusive):
   * Print '\*' repeated i times

**Pattern 2: Decreasing Asterisks**

1. Print a heading: "Pattern 2"
2. Repeat for i from rows down to 1:
   * Print '\*' repeated i times

**Pattern 3: Increasing Number Triangle**

1. Print a heading: "Pattern 3"
2. Repeat for i from 1 to rows (inclusive):
   * For each j from 1 to i:
     + Print j on the same line with a space
   * Print a newline after each row

**Pattern 4: Decreasing Number Triangle**

1. Print a heading: "Pattern 4"
2. Repeat for i from rows down to 1:
   * For each j from 1 to i:
     + Print j on the same line with a space
   * Print a newline after each row

**11. End**

***Print '\*' repeated i times***

Loop from i = 4 to 1:

**Output: "Pattern 2"**

***Print '\*' repeated i times***

*False*

Loop from i = 1 to 4:

**Output: "Pattern 1"**

Set rows ← 4

Loop from i = 1 to 4:

***Print '\*' repeated i times***

|  |  |
| --- | --- |
| Program 6: Write a program to print Patterns. | Chapter-4 Control Statements  Date  Page No. |
|  |  |
|  | def main():  # Input number of rows from user  rows = 4  # Loop to print pattern  print( "\n\n\n\n Pattren 1 \n ")  for i in range(1, rows + 1):  print('\*' \* i) #This line prints a string of asterisks (\*), repeated i times.  #In Python, 'string' \* number repeats the string.  print("\n\n\n\n Pattren 2 \n ")  for i in range(rows,0,-1):  print('\*' \* i) #This line prints a string of asterisks (\*), repeated i times.    print("\n\n\n\n Pattren 3 \n ")  # Outer loop for rows  for i in range(1, rows + 1):  # Inner loop for numbers in each row  for j in range(1, i + 1):  print(j, end=' ')  print() # Newline after each row  print( "\n\n\n\n Pattren 4 \n ")  # Outer loop for rows  for i in range(rows,0,-1):  # Inner loop for numbers in each row  for j in range(1, i + 1):  print(j, end=' ')  print() # Newline after each row  # Call the main function  if \_\_name\_\_ == "\_\_main\_\_":  main() |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |