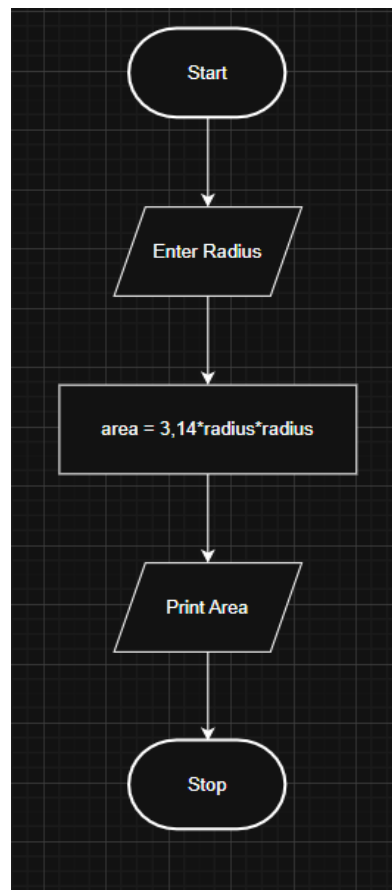


1.1.1 Area of circle

Algorithm:

1. Start.
2. Declare variables radius, area.
3. Set the value of pi = 3.14.
4. Read the Radius from the user.
5. Calculate the area using the formula ($\text{area} = 3.14 * \text{radius} * \text{radius}$).
6. Display the calculated area.
7. Stop.

Flowchart:



Code:

```
radius = float(input())  
area = 3.14 * radius * radius  
print(f"{area:.4f}")
```

Execution:

CODETANTRA

Home

kartik.jaurkar.batch2025@sitnagpur.siu.edu.inSupportLogout

1.1.1. Area of Circle

20.08

AA

Write a Python program that calculates the area of a circle when the radius is provided by the user. Use $\pi = 3.14$ and display the area.

Input Format:

- A single line containing a floating-point number representing the radius.

Output Format:

- Print the computed area of the circle formatted to 4 decimal places.

Sample Test Cases

circlearea...

Submit

Debugger

1radius = float(input ())

2area = 3.14 * radius *radius

3print(f"area:.4f")

Average time

0.010 s

9.75 ms

Maximum time

0.015 s

15.00 ms

2 out of 2 shown test case(s) passed

2 out of 2 hidden test case(s) passed

Test case 115 ms

Test case 210 ms

Terminal

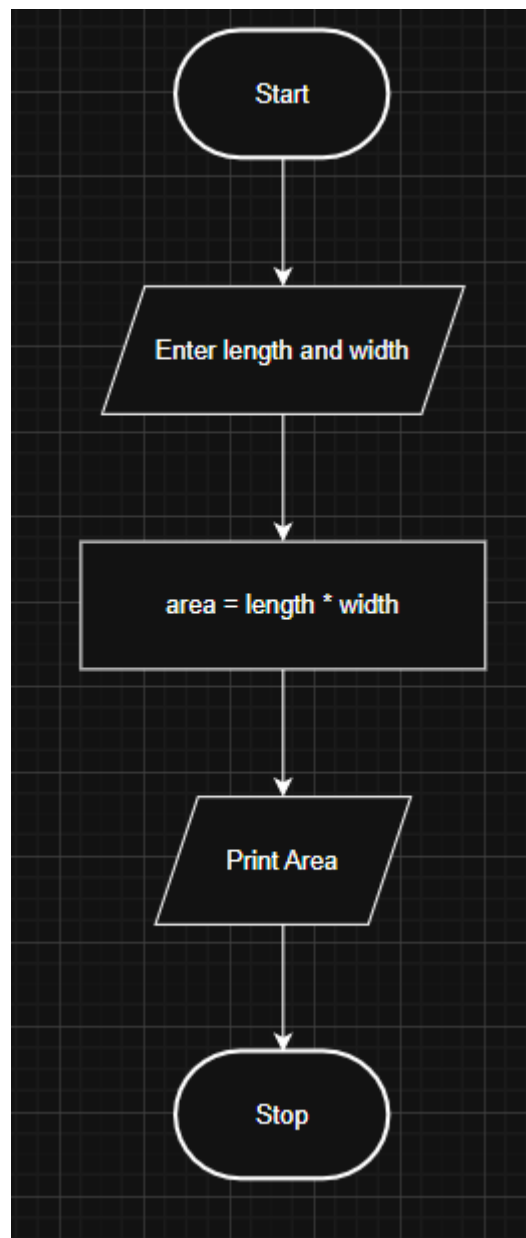
Test cases

1.1.2Area of Rectangle :

Algorithm:

1. Start.
2. Read the length and width of rectangle from the user.
3. Calculate the area using the formula($\text{Area} = \text{length} * \text{width}$).
4. Display the calculated Area.
5. Stop.

Flowchart:



Code:

```
length = float(input())  
width = float(input())  
  
area = length * width  
print(f"{area:.2f}")
```

Execution:

The screenshot displays the CODETANTRA IDE interface. On the left, a panel titled "1.1.2. Area of Rectangle" provides instructions and formulas for calculating the area of a rectangle. The main editor on the right shows the Python code for this task. Below the code editor, a "Test cases" panel indicates that all 5 test cases (both shown and hidden) have passed. The bottom of the interface features navigation buttons: "Prev", "Reset", "Submit", and "Next".

CODETANTRA Home

Write a Python program to calculate the area of a rectangle given its length and width.

Formula:
Area of Rectangle = Length × Width

Input Format:

- First line contains a float value representing the length of the rectangle
- Second line contains a float value representing the width of the rectangle

Output Format:

- Print the area of the rectangle as a float value formatted to 2 decimal places.

Sample Test Cases

areaOfRe...

```
1 length = float(input())  
2 width = float(input())  
3  
4 area = length * width  
5 print(f"{area:.2f}")
```

Average time: 0.011 s
Maximum time: 0.042 s

5 out of 5 shown test case(s) passed
5 out of 5 hidden test case(s) passed

- Test case 1: 42 ms
- Test case 2: 8 ms
- Test case 3: 10 ms
- Test case 4: 6 ms

Terminal Test cases

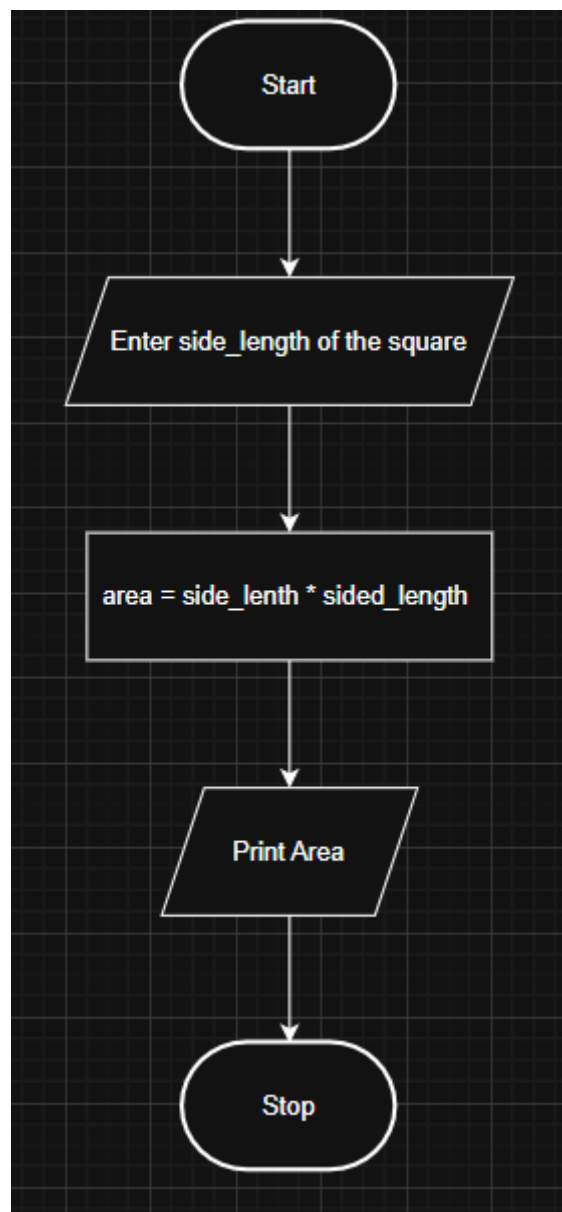
< Prev Reset Submit Next >

1.1.3 Calculate Area of Square :

Algorithm:

1. Start.
2. Declare variable side_length.
3. Read the value of the side_length from the user.
4. Calculate the area using the formula ($\text{area} = \text{side_length} * \text{side_length}$).
5. Display the calculated area.
6. Stop.

Flowchart:



Code:

```
side_length = int(input())  
area = side_length * side_length  
print(area)
```

Execution:

The screenshot displays the CODETANTRA web-based IDE interface. On the left, a panel titled "1.1.3. Calculate Area of the Square" provides instructions: "Write a Python program that prompts the user to enter the *side_length* of a square and computes the area of the square." It also lists the formula $\text{Area} = \text{side_length}^2$, the input format (a positive integer for *side_length*), and the output format (a positive integer for the area). Below this is a "Sample Test Cases" section with a "+" icon.

The main editor on the right shows the following Python code:

```
1 side_length = int(input())  
2 area = side_length * side_length  
3  
4 print(area)  
5  
6  
7  
8  
9  
10
```

Below the code editor, the execution results are shown. The "Average time" is 0.015 s (15.00 ms) and the "Maximum time" is 0.021 s (21.00 ms). The status indicates "2 out of 2 shown test case(s) passed" and "2 out of 2 hidden test case(s) passed". A list of test cases is displayed:

Test Case	Time	Status
Test case 1	15 ms	Passed
Test case 2	14 ms	Passed

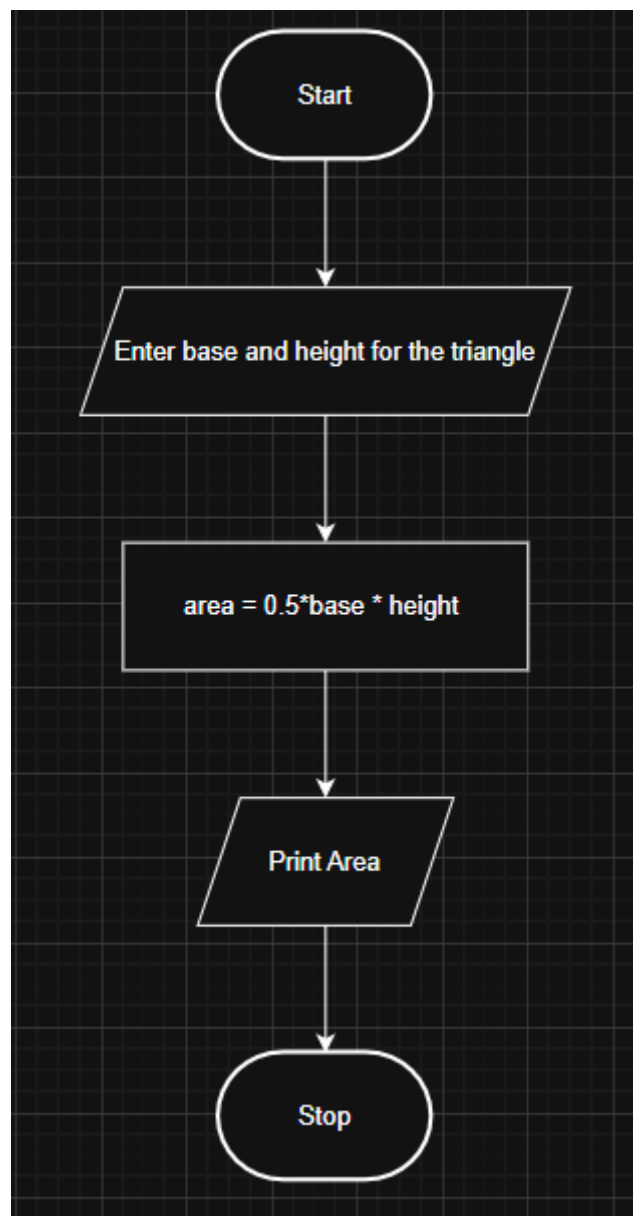
At the bottom, there are tabs for "Terminal" and "Test cases".

1.1.4 Area of Triangle:

Algorithm:

1. Start.
2. Declare the variables base , height , area.
3. Read the value of base and height from the user.
4. Calculate the area using the formula ($\text{area} = 0.5 * \text{base} * \text{height}$).
5. Display the Area
6. Stop.

Flowchart:



Code:

```
base = float(input())  
height = float(input())  
area = 0.5 * base * height  
print(f"{area:.2f}")
```

Execution:

The screenshot displays the CODETANTRA web-based IDE interface. On the left, the problem statement for "1.1.4. Area of Triangle" is shown, including the formula $\text{Area of Triangle} = 0.5 \times \text{base} \times \text{height}$, input/output formats, and sample test cases. The main editor on the right contains the Python code for calculating the area. Below the code editor, the execution results are displayed, showing that 2 out of 2 shown test cases and 2 out of 2 hidden test cases passed. The average time for the execution is 0.018 s (18.00 ms) and the maximum time is 0.027 s (27.00 ms). The test cases are listed as Test case 1 (27 ms) and Test case 2 (14 ms). The interface also includes a terminal, a submit button, and navigation controls at the bottom.

1.1.4. Area of Triangle 01:55

Write a Python program that prompts the user to enter the triangle's base and height and computes the triangle's area.

Formula: $\text{Area of Triangle} = 0.5 \times \text{base} \times \text{height}$.

Input Format:

- The first line of input is the float value that represents the base of the triangle.
- The second line of input is the float value that represents the height of the triangle.

Output Format:

- The output is the floating point value that represents the area of a triangle, formatted to two decimals.

Sample Test Cases +

triangleA... Submit

```
1 base = float(input())  
2 height = float(input())  
3 area = 0.5 * base * height  
4  
5 print(f"{area:.2f}")
```

Average time: **0.018 s** 18.00 ms Maximum time: **0.027 s** 27.00 ms

2 out of 2 shown test case(s) passed
2 out of 2 hidden test case(s) passed

Test case 1 27 ms
Test case 2 14 ms

Terminal Test cases

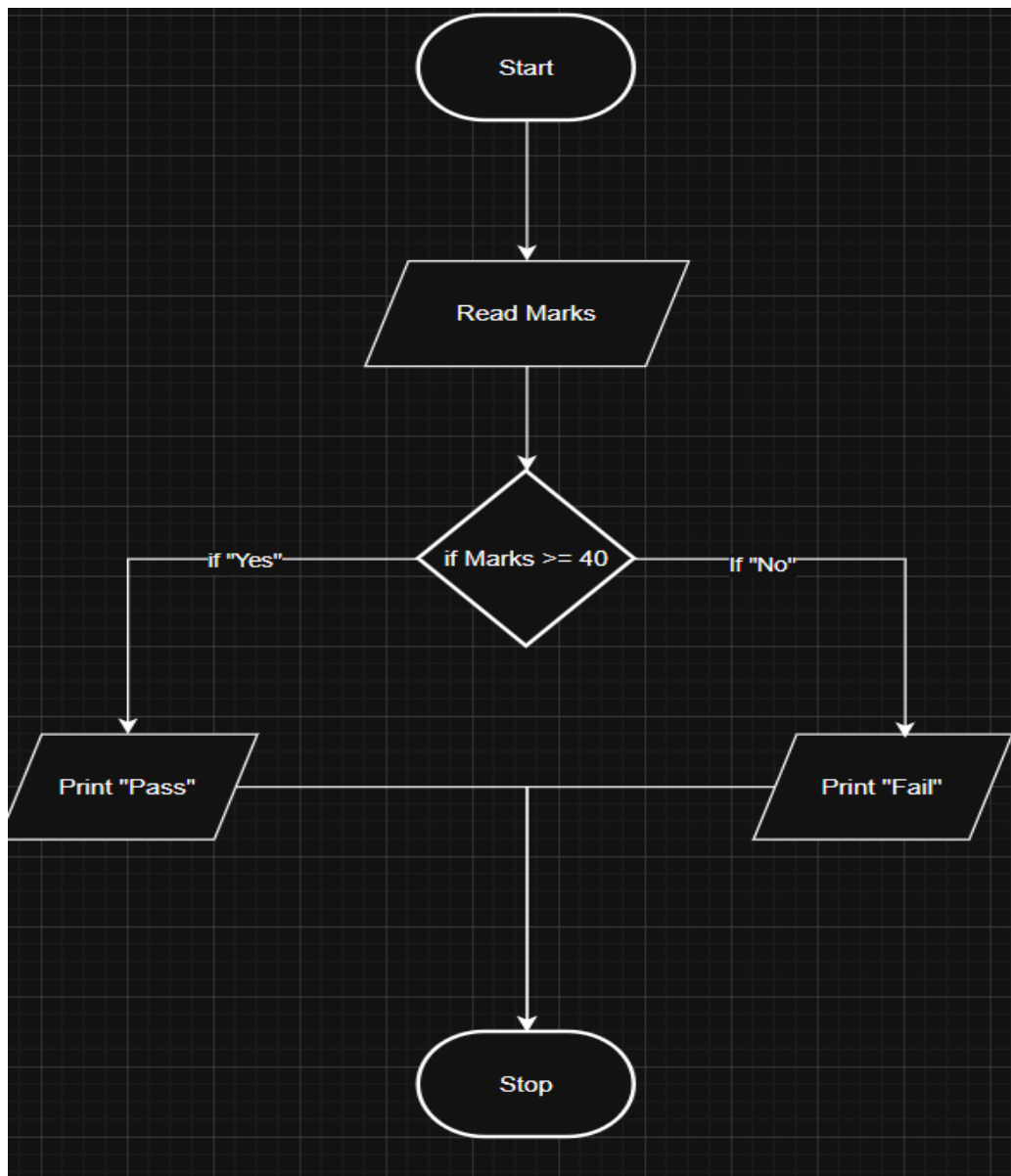
< Prev Reset Submit Next >

1.1.5 Student pass or fail Status:

Algorithm:

1. Start.
2. Declare variables Marks.
3. Read the marks obtained by the student .
4. Check condition :
 - If marks ≥ 40 , then print “Pass”.
 - Else print “Fail”
5. Stop,

Flowchart:



Code :

```
marks = int(input())
```

```
if marks >= 40:
```

```
    print("Pass")
```

```
else:
```

```
    print("Fail")
```

Execution:

The screenshot displays the CodeTANTRA IDE interface. On the left, the problem statement '1.1.5. Student Pass or Fail Status' is shown, along with the pass/fail criteria and input/output formats. The main editor on the right contains the Python code for the program. Below the code, the execution results are displayed, showing that all test cases passed.

CodeTANTRA Home kartik.jaurkar.batch2025@sitnagpur.siu.edu.in Support Logout

1.1.5. Student Pass or Fail Status 03:20

Write a Python program to determine whether a student passed the exam or not based on their marks.

Pass/Fail Criteria:

- A student passes if marks ≥ 40
- A student fails if marks < 40

Input Format:

- Single line contains an integer representing the marks obtained by the student.

Output Format:

- Print "Pass" if the student passed the exam.
- Print "Fail" if the student failed the exam.

Sample Test Cases +

passOrFa... Submit

```
1 marks = int(input())
2
3 if marks >= 40:
4     print("Pass")
5 else:
6     print("Fail")
```

Average time: 0.010 s (9.86 ms) Maximum time: 0.016 s (16.00 ms)

3 out of 3 shown test case(s) passed
4 out of 4 hidden test case(s) passed

Test case 1: 15 ms
Test case 2: 15 ms
Test case 3: 9 ms

Terminal Test cases

< Prev Reset Submit Next >