

## 1.1.1 Area of Circle

- Algorithm

STEP 1 : Start

STEP 2 : Input radius

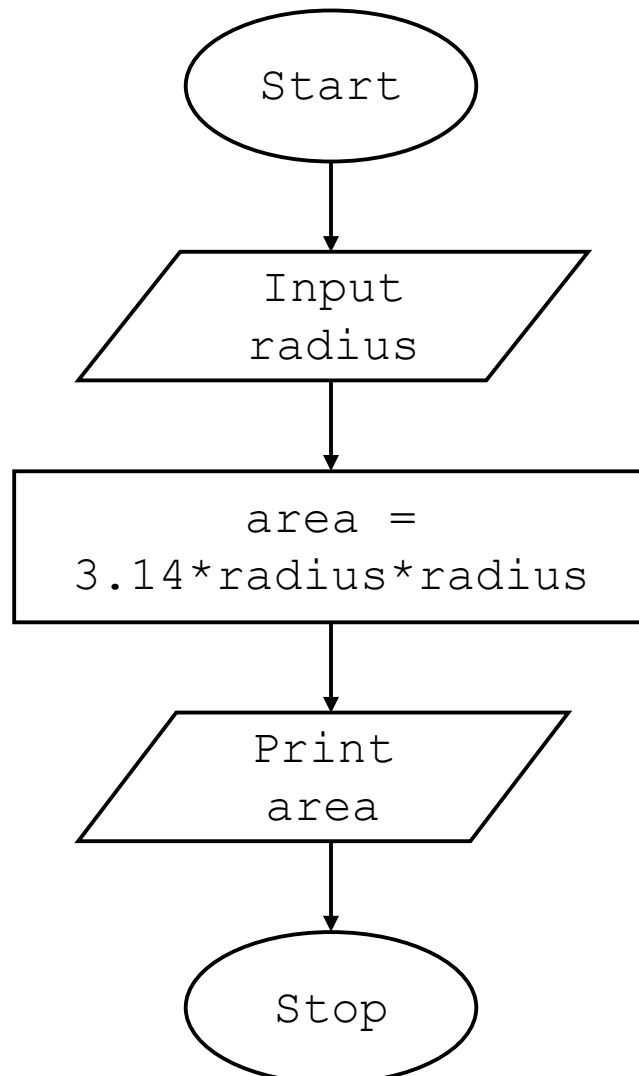
STEP 3 : Calculate

$\text{area} = 3.14 * \text{radius} * \text{radius}$

STEP 4 : Print area

STEP 5 : Stop

- Flowchart



- Code

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

- Execution

The screenshot displays the CODETANTRA online IDE interface. On the left, the problem statement for "1.1.1. Area of Circle" is shown, requiring a Python program to calculate the area of a circle given its radius. The input format is a single line with a floating-point number, and the output format is the area formatted to 4 decimal places. Below the problem statement is a section for "Sample Test Cases".

The main editor area shows the following Python code:

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

Below the code editor, the execution results are displayed. The "Average time" is 0.083 s (82.75 ms) and the "Maximum time" is 0.203 s (203.00 ms). The status indicates "2 out of 2 shown test case(s) passed" and "2 out of 2 hidden test case(s) passed".

Test case details:

Test Case	Expected output	Actual output
Test case 1	35.4493	35.4493
Test case 2	35.4493	35.4493

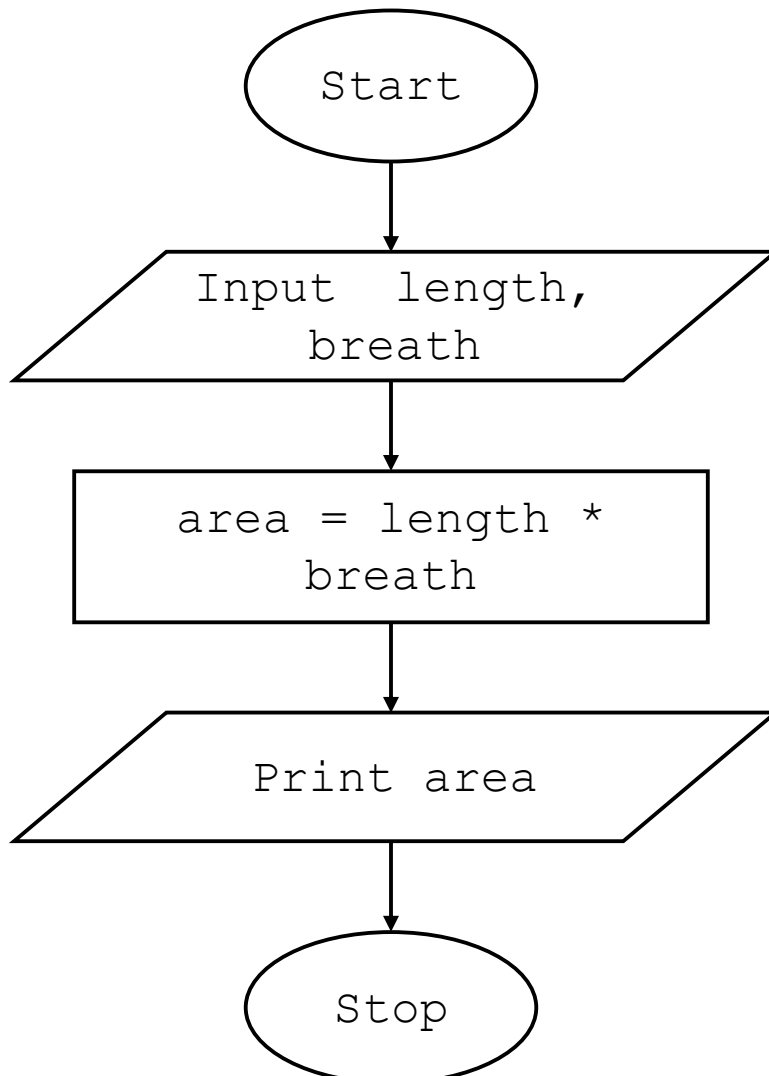
At the bottom of the interface, there are buttons for "Terminal", "Test cases", "Prev", "Reset", "Submit", and "Next".

## 1.1.2 Area of Rectangle

- Algorithm

```
STEP 1 : Start
STEP 2 : Input length, breath
STEP 3 : Calculate
          area = length*breath
STEP 4 : Print area
STEP 5 : Stop
```

- Flowchart



- Code

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

- Execution

The screenshot displays the CODETANTRA online IDE interface. On the left, a sidebar contains the problem description for "1.1.2. Area of Rectangle". It includes the instruction "Write a Python program to calculate the area of a rectangle given its length and width.", the formula "Area of Rectangle = Length × Width", the input format (two lines of float values), and the output format (a float value formatted to 2 decimal places). Below this is a section for "Sample Test Cases".

The main editor area shows a Python script with four lines of code:

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

Below the code editor, the execution results are displayed. It shows the average time (0.028 s) and maximum time (0.048 s) for the program. The results indicate that "5 out of 5 shown test case(s) passed" and "5 out of 5 hidden test case(s) passed". A detailed view of "Test case 1" is shown, comparing the expected output (18.3, 3.2, 54.60) with the actual output (18.3, 3.2, 54.60). The "Test case 2" is also shown as passed.

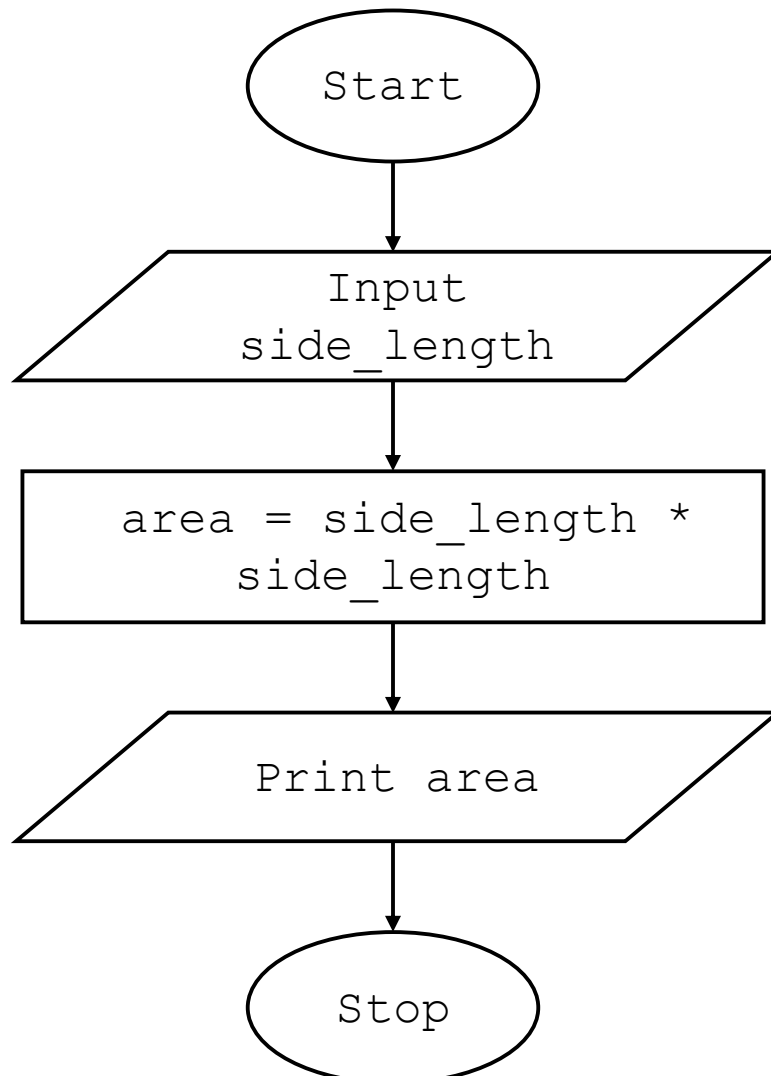
At the bottom of the interface, there are buttons for "Terminal", "Test cases", "Prev", "Reset", "Submit", and "Next".

## 1.1.3 Calculate Area of the Square

- Algorithm

```
STEP 1 : Start
STEP 2 : Input side_length
STEP 3 : Calculate
          area = side_length * side_length
STEP 4 : Print area
STEP 5 : Stop
```

- Flowchart



- Code

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

- Execution

The screenshot displays the CODETANTRA online IDE interface. On the left, the problem statement for '1.1.3. Calculate Area of the Square' is shown, including the formula  $\text{Area} = \text{side\_length}^2$  and input/output specifications. The main editor shows the Python code: `side_length = int(input())`, `area = side_length * side_length`, and `print(area)`. The right sidebar shows the execution results, indicating that 2 out of 2 test cases passed. The first test case shows an expected output of 25 and an actual output of 25. The bottom of the interface includes buttons for 'Terminal', 'Test cases', 'Prev', 'Reset', 'Submit', and 'Next'.

CODETANTRA Home

1.1.3. Calculate Area of the Square

Write a Python program that prompts the user to enter the *side\_length* of a square and computes the area of the square.

Formula:

- Area =  $\text{side\_length}^2$

Input Format:

- The input is a positive integer value that represents the *side\_length* of the square.

Output Format:

- The output is a positive integer value that represents the area of the square.

Sample Test Cases

AreaSqua...

```
1 side_length = int(input())  
2 area = side_length * side_length  
3 print(area)  
4  
5
```

Average time: 0.025 s  
Maximum time: 0.056 s

25.50 ms 56.00 ms

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

Test case 1 56 ms

Expected output: 25  
Actual output: 25

Test case 2 0 ms

Terminal Test cases

< Prev Reset Submit Next >

## 1.1.4 Area of Triangle

- Algorithm

STEP 1 : Start

STEP 2 : Input base, height

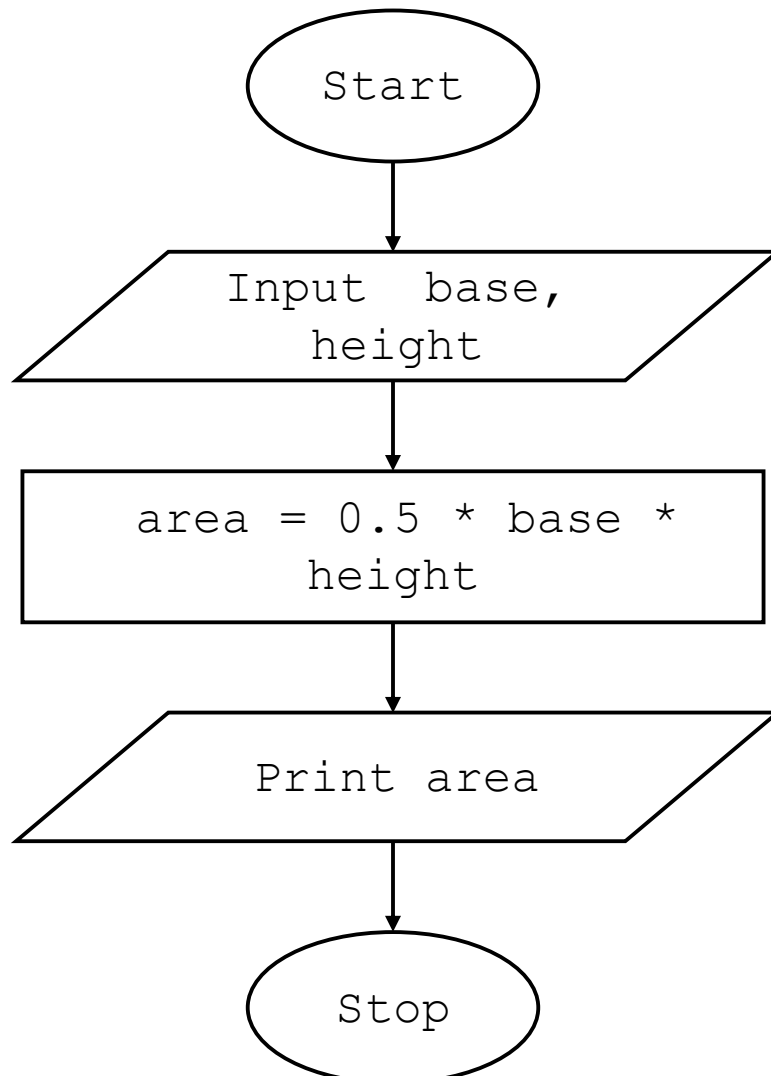
STEP 3 : Calculate

$\text{area} = 0.5 * \text{base} * \text{height}$

STEP 4 : Print area

STEP 5 : Stop

- Flowchart



- Code

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

- Execution

**CODETANTRA** Home

1.1.4. Area of Triangle

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...

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

Average time: 0.049 s  
Maximum time: 0.096 s

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

Test case 1 96 ms

Expected output	Actual output
6.56	6.56
1.23	1.23
4.82	4.82

Test case 2 25 ms

Terminal Test cases

< Prev Reset Submit Next >



## 1.1.5 Student Pass or Fail status

- Algorithm

STEP 1 : Start

STEP 2 : Input marks

STEP 3 : Check condition

    If marks  $\geq 40$

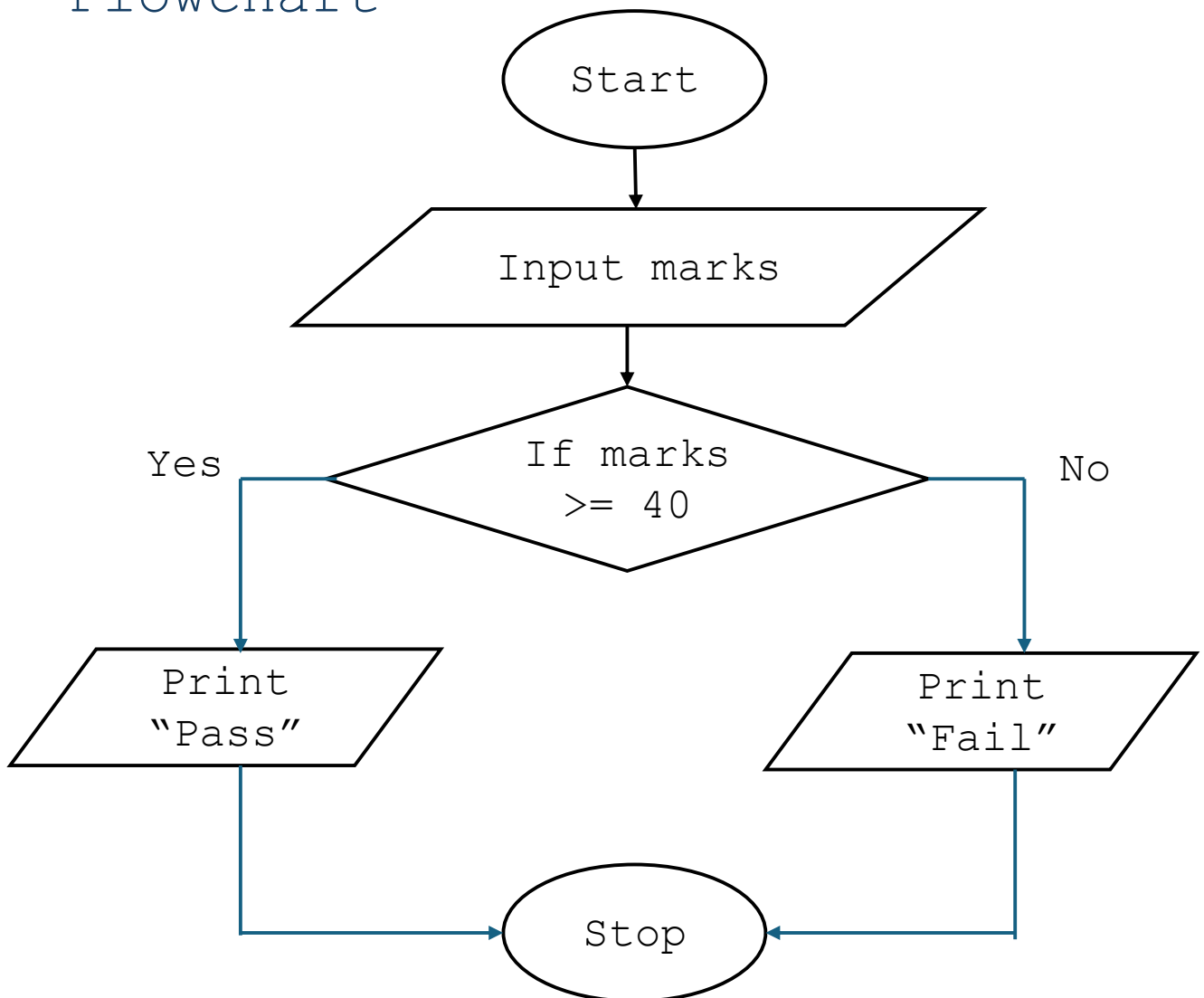
        Print "Pass"

    Else

        Print "Fail"

STEP 4 : Stop

- Flowchart



- Code

```
marks=int(input())
if marks>= 40:
print("Pass")
else:
print("Fail")
```

- Execution

**CODETANTRA** Home

1.1.5. Student Pass or Fail Status

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  $\geq 40$
- 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...

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

Average time: 0.015 s (15.29 ms) | Maximum time: 0.033 s (33.00 ms)

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

Test case 1 (3 ms)

Expected output	Actual output
45	45
Pass	Pass

Test case 2 (12 ms)

Test case 3 (40 ms)

Terminal | Test cases

< Prev | Reset | Submit | Next >