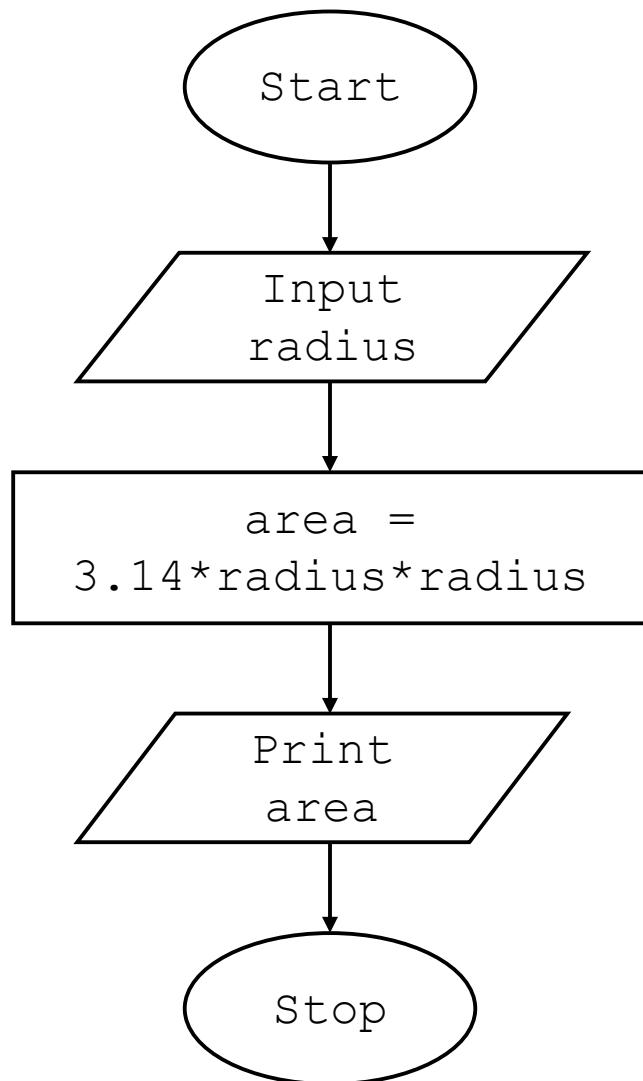


1.1.1 Area of Circle

- 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 shows the CodeTantra IDE interface. On the left, there's a sidebar with sections for '1.1.1. Area of Circle', 'Input Format' (a single line float), and 'Output Format' (4 decimal places). The main workspace contains a code editor with the following Python script:

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

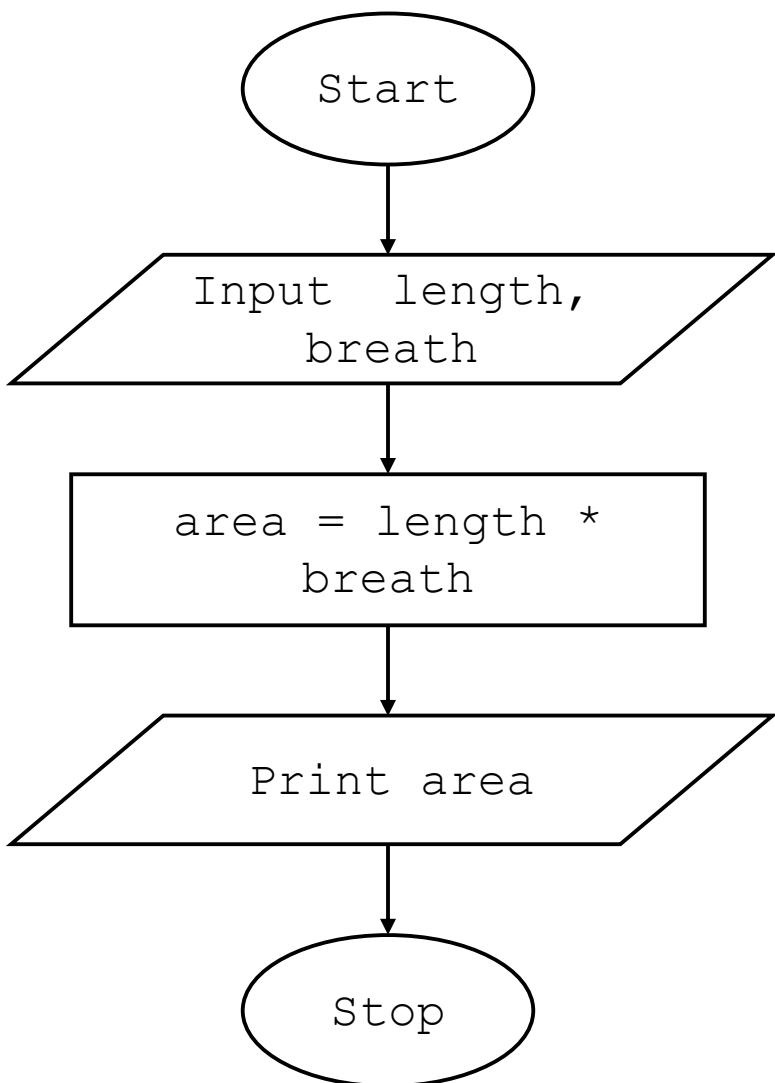
Below the code editor, performance metrics are displayed: Average time 0.003 s, Maximum time 0.203 s, and two test cases both passed (2 out of 2 shown and 2 out of 2 hidden). The test cases show expected output 35.4493 and actual output 35.4493 for both cases. At the bottom, there are tabs for Terminal and Test Cases, along with navigation buttons for Prev, Reset, Submit, and Next.

1.1.2 Area of Rectangle

- Algorithm

STEP 1 : Start
STEP 2 : Input length, breath
STEP 3 : Calculate
 $\text{area} = \text{length} * \text{breath}$
STEP 4 : Print area
STEP 5 : Stop

- Flowchart



- Code

```

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

```

- Execution

CODETANTRA Home

aryan.kamdi.batch2025@nitagpur.siu.edu.in * Support Logout

1.1.2. Area of Rectangle

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

areaOfRect.py

```

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

```

Average time: 0.028 s Minimum time: 0.048 s
27.00 ms 48.00 ms

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

Test case 1	Expected output	Actual output
18.5 5.2	95.5	95.5
54.00	54.00	54.00

Test case 2	Expected output	Actual output
10.0 2.0	20.0	20.0
20.00	20.00	20.00

Terminal Test cases

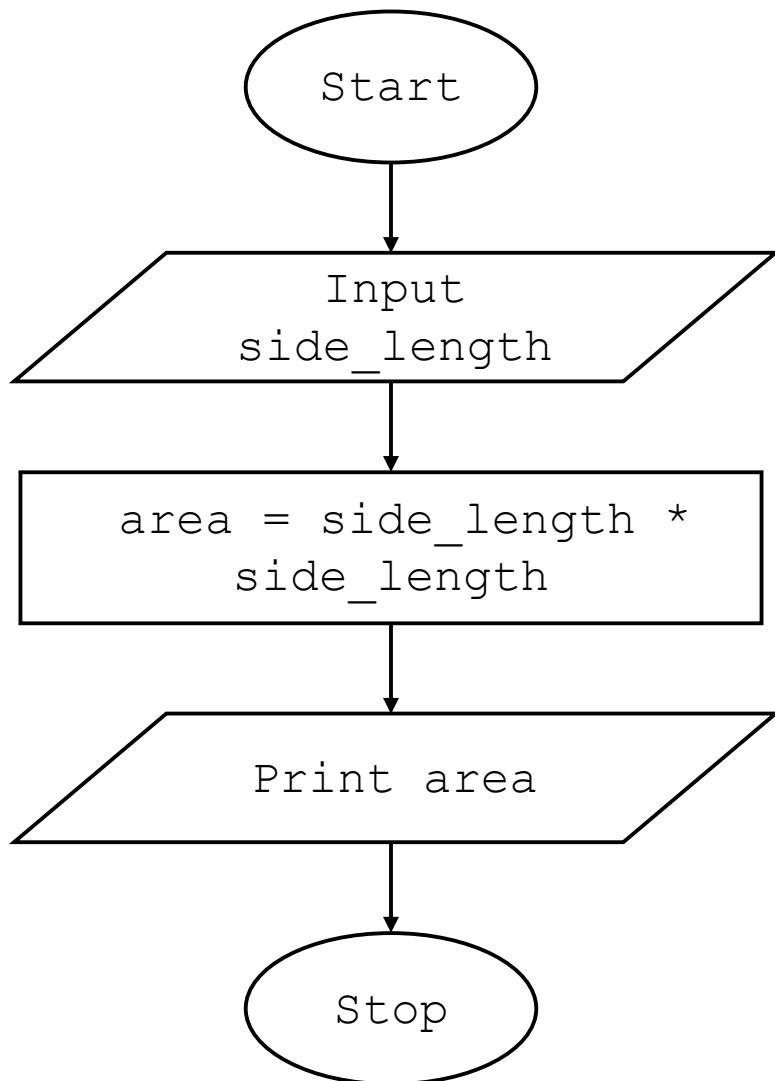
< Prev Reset Submit Next >

1.1.3 Calculate Area of the Square

• Algorithm

STEP 1 : Start
STEP 2 : Input side_length
STEP 3 : Calculate
 $\text{area} = \text{side_length} * \text{side_length}$
STEP 4 : Print area
STEP 5 : Stop

• Flowchart



- Code

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

- Execution

The screenshot shows the CodeTantra IDE interface. The title bar says "CODETANTRA Home". The user is logged in as "aryan.kamdi.batch2025@strnagpururu.edu.in". The code editor window is titled "AreaSqua...". It contains the following Python code:

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

Below the code editor, performance metrics are displayed: Average time 0.025 s, Maximum time 0.056 s, 25.50 ms, and 56.00 ms. To the right, test results show "2 out of 2 shown test case(s) passed" and "2 out of 2 hidden test case(s) passed". Two test cases are listed:

- Test case 1**: Expected output 25, Actual output 25.
- Test case 2**: Expected output 100, Actual output 100.

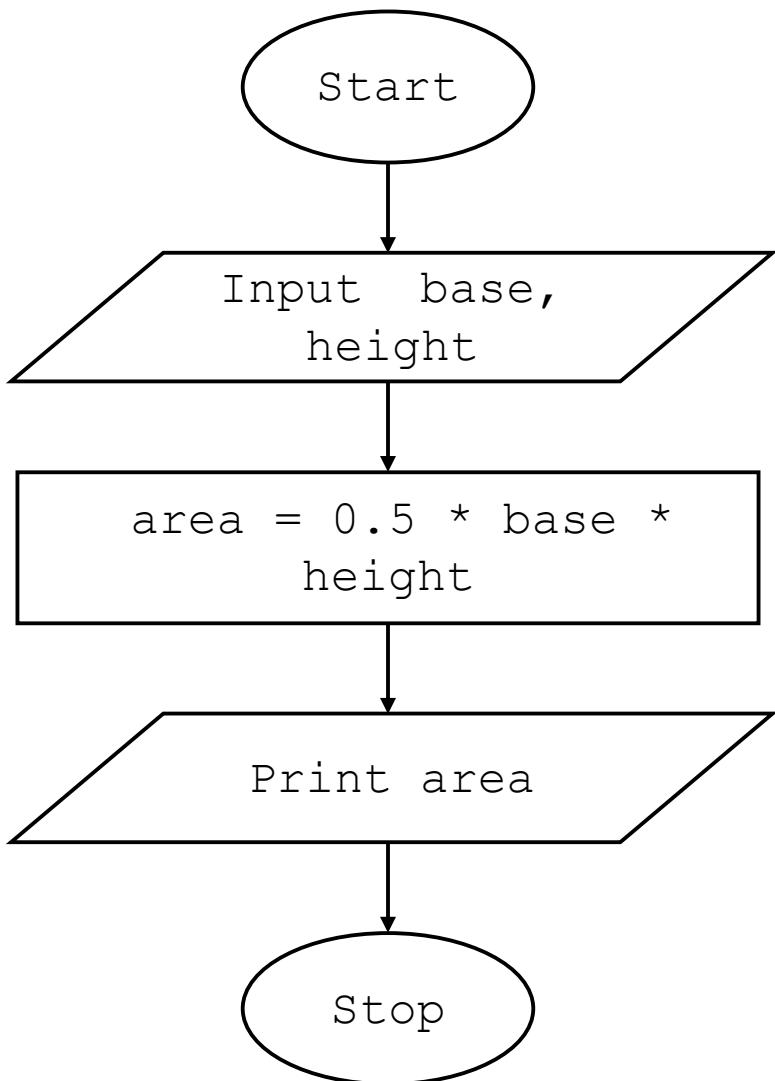
At the bottom, there are buttons for "Terminal", "Test cases", and navigation links: < Prev, Reset, Submit, Next >.

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

The screenshot shows the CodeTantra IDE interface. On the left, there's a problem statement for "1.1.4. Area of Triangle". It asks to write a Python program that prompts the user for base and height and prints the area. A formula $Area = 0.5 \times base \times height$ is provided. Below it, "Input Format" and "Output Format" sections are listed with bullet points. On the right, the code editor shows a file named "triangleA..." with the following code:

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

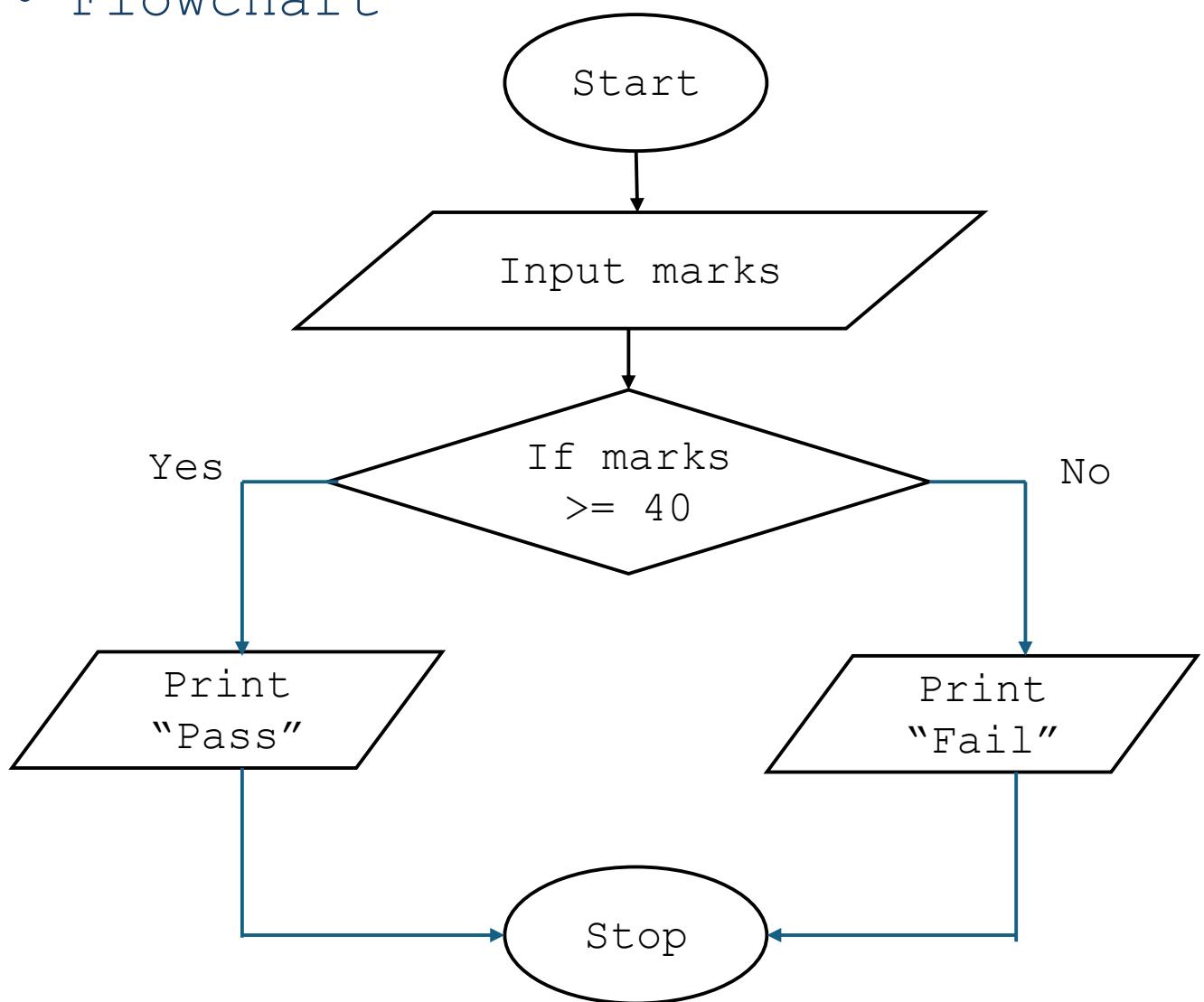
The code editor includes performance metrics: Average time 0.049 s, Maximum time 0.096 s, and two test cases both marked as passed. At the bottom, there are tabs for Terminal and Test cases, and buttons for Prev, Reset, Next, and Run.

1.1.5 Student Pass or Fail status

• Algorithm

STEP 1 : Start
STEP 2 : Input marks
STEP 3 : Check condition
 If marks ≥ 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

ayan.kamdi.batch2025@sitnagpur.saiu.edu.in Support Logout

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 ≥ 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

passOrFail.py

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

Average time: 0.015 s Maximum time: 0.033 s
15.29 ms 33.00 ms

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

Test case 1	33 ms	Debug
Expected output	Actual output	
45	45	
Pass	Pass	

Test case 2	32 ms
Test case 3	33 ms

Terminal Test cases

< Prev Reset Submit Next >