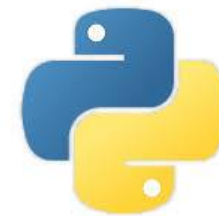


CMPT 120: Introduction to Computing Science and Programming 1

Procedural programming in Python



python™

Copyright © 2018, Liaqat Ali. Based on [CMPT 120 Study Guide](#) and [Think Python - How to Think Like a Computer Scientist](#), mainly. Some content may have been adapted from earlier course offerings by Diana Cukierman, Anne Lavergn, and Angelica Lim. Copyrights © to respective instructors. Icons copyright © to their respective owners.

One-Stop Access To Course Information

- **Course website**: One-stop access to all course information.

<http://www2.cs.sfu.ca/CourseCentral/120/liaqata/WebSite/index.html>

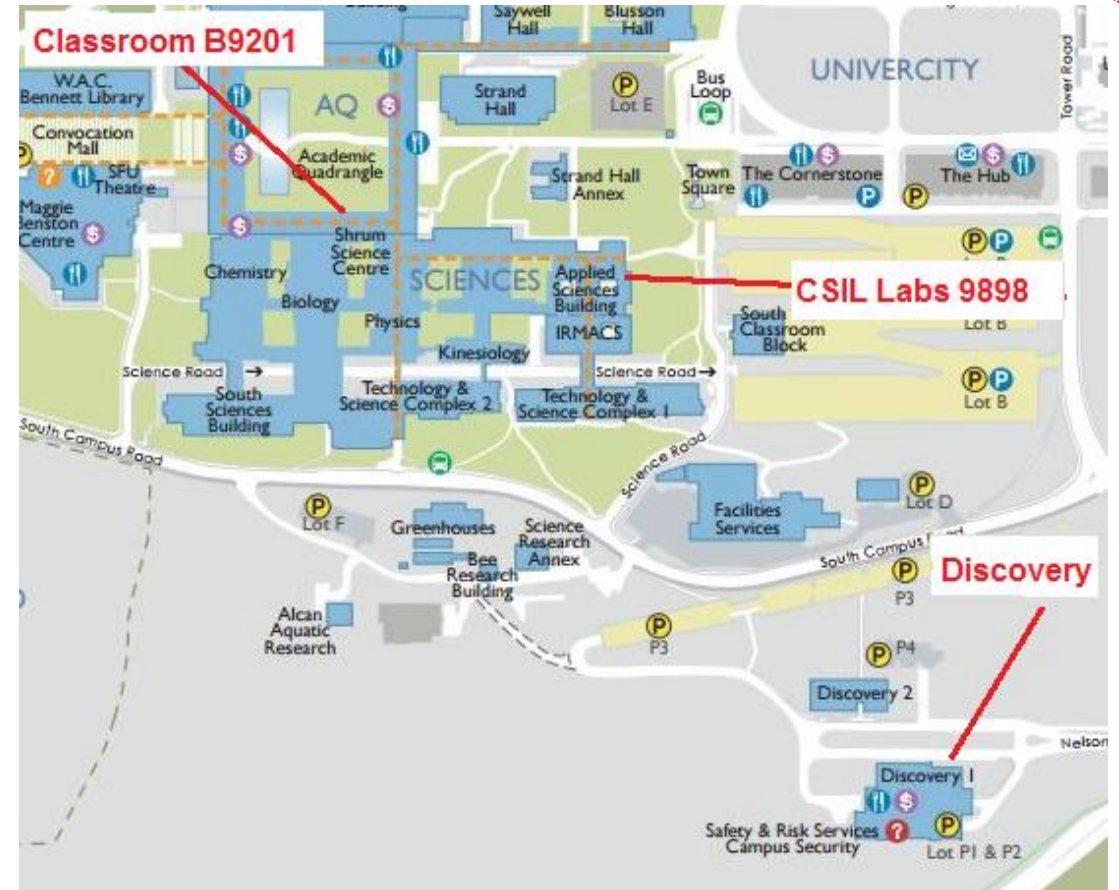
- Course Outline
- Exam Schedule
- Python Info
- CourSys/Canvas link
- Learning Outcomes
- Office Hours
- Textbook links
- and more...
- Grading Scheme
- Lab/Tutorial Info
- Assignments

- **Canvas**: Discussions forum - <https://canvas.sfu.ca/courses/39187>

- **CourSys**: Assignments submission, grades - www.coursys.sfu.ca

Some Reminders

- **Get familiar with the course Website.**
 - <http://www2.cs.sfu.ca/CourseCentral/120/liaqata/WebSite/index.html>
 - Minor updates may occur during first week.
- **Get fob to access LABS (start next week!)**
 - If you don't have it already, get a new fob from **Discovery Park 1**.



Additional Resources / Online References

- Online references are **as important as the texts**. (Links on course website.)
- These resources are **very important to your success**.
 - They aren't meant to be read from beginning to end like the readings in the textbook.
- You should **use them to get an overall picture of the topic** and as references as you do the assignments.

How to Learn in This Course?



- A** **Attend** Lectures & Labs
- R** **Read** / review Textbook/Slides/Notes
- R** **Reflect** and ask Questions
- O** **Organize** – your learning activities on weekly basis,
and finally...
- W** **Write** Code, **Write Code**, and **Write Code**.

Course Topics

1. General introduction
2. Algorithms, flow charts and pseudocode
3. **Procedural programming in Python**
4. Data types and control structures
5. Fundamental algorithms
6. Binary encodings
7. Basics of computability and complexity
8. Basics of Recursion
9. Subject to time availability:
 - Basics of Data File management

Today's Topics

1. Procedural Programming in Python

- Transforming an algorithm to a program
- Write Code, Write Code, and Write Code.

Today's Topics

1

**Transforming an algorithm to a program:
Write Code, Write Code, and Write Code**

Write a Python Program to Add Two Numbers

Step 1: Start

Step 2: Declare a variable n1 and initialize it to 0.

Step 3: Declare a variable n2 and initialize it to 0.

Step 4: Declare a variable sum and initialize it to 0.

Step 5: Get 1st number from user and store in n1.

Step 6: Get 2nd number from user and store in n2.

Step 7: Add n1 and n2, and store the answer in sum.

Step 8: Display SUM

Step 9: End

Write a Python Program to Add Two Numbers

Step 1: Start

Step 2: Declare a variable n1 and initialize it to 0.

Step 3: Declare a variable n2 and initialize it to 0.

Step 4: Declare a variable sum and initialize it to 0.

Step 5: Get 1st number from user and store in n1.

Step 6: Get 2nd number from user and store in n2.

Step 7: Add n1 and n2, and store the answer in sum.

int

int

Step 8: Display SUM

Step 9: End

Write a Python Program to Add Three Numbers

Step 1: Start

Step 2: Declare a variable n1 and initialize it to 0.

Step 3: Declare a variable n2 and initialize it to 0.

Step 4: Declare a variable n3 and initialize it to 0.

Step 5: Declare a variable sum and initialize it to 0.

Step 6: Get 1st number from user and store in n1.

Step 7: Get 2nd number from user and store in n2.

Step 8: Get 2nd number from user and store in n2.

Step 9: Add n1, n2, n3 and assign the result to SUM.

$$\text{SUM} \leftarrow n1 + n2 + n3$$

Step 10: Display SUM

Step 11: End

A yellow rectangular box containing the text 'int' in white lowercase letters.A yellow rectangular box containing the text 'int' in white lowercase letters.A yellow rectangular box containing the text 'int' in white lowercase letters.

Write a Python Program to Solve $2x+2y$

Step 1: Start

Step 2: Declare a variable `x` and initialize it to 0.

Step 3: Declare a variable `y` and initialize it to 0.

Step 4: Declare a variable `ans` and initialize it to 0.

Step 5: Get value of `x` from user and store in `x`.

Step 6: Get value of `y` from user and store in `y`.

Step 7:

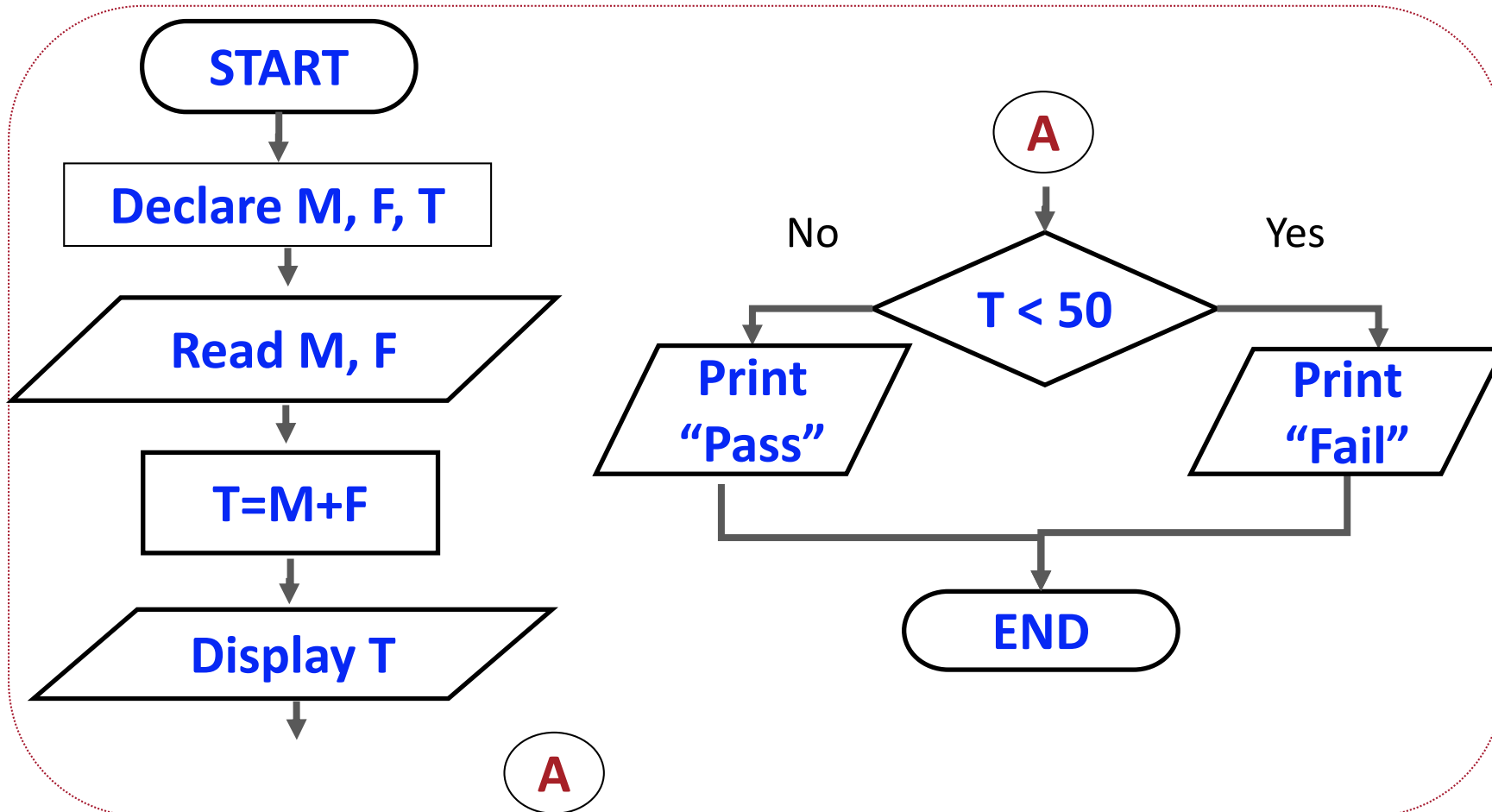
Step 8: Display SUM

Step 9: End

`int`

`int`

Write a Program for Following Marks Flowchart



int

int

⋮

⋮

Program: Find the Smaller of Two Numbers

Step 1: Start

Step 2: Declare variable n1 to store the 1st number.

Step 3: Declare variable n2 to store the 2nd number.

Step 4: Get the value of n1 from the user.

Step 5: Get the value of n2 from the user.

Step 6: If $n1 < n2$ then print n1.

else print n2.

Step 8: End

Program: Find the Smallest of Three Numbers

Read n_1 , n_2 , n_3 .

If $n_1 < n_2$ and $n_1 < n_3$,

Write n_1 .

If $n_2 < n_1$ and $n_2 < n_3$,

Write n_2 .

If $n_3 < n_1$ and $n_3 < n_2$,

Write n_3 .

Program: Find Sum of First 100 Natural Numbers

Set sum to 0

Set n to 1

Repeat until $n \leq 100$:

Set $sum = sum + n$

$n = n + 1$

Write S

Algorithm: Convert Height In Meters To Feet and Inches

- 1: Start**
- 2: Declare meter, feet, total inches and inches variables.**
- 3: Initialize feet, total inches and inches variables to 0.**
- 4: Get the height in meters from the user.**
- 5: Convert meters into total inches and store it.**
- 5: Convert total inches into feet and store it.**
- 6: Find remainder of total inches / 12 and store in inches.**
- 7: Display the value in feet variable.**
- 8: Display the value in the inches variable.**

whole number

remainder

9: End

Liaqat Ali, Summer 2018.

Program: Convert Height In Meters To Feet and Inches

Read *meters*

Set *total_inch* to $39.37 \times \text{metres}$

Set *feet* to **whole number** part of $\text{totalInch} / 12$

Set *inches* to **remainder** of $\text{totalInch} / 12$

Write *feet, inches*





Questions?