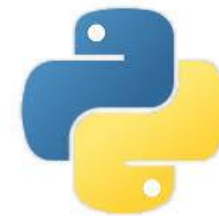


CMPT 120: Introduction to Computing Science and Programming 1

Algorithms, Flowcharts and Pseudocodes



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>

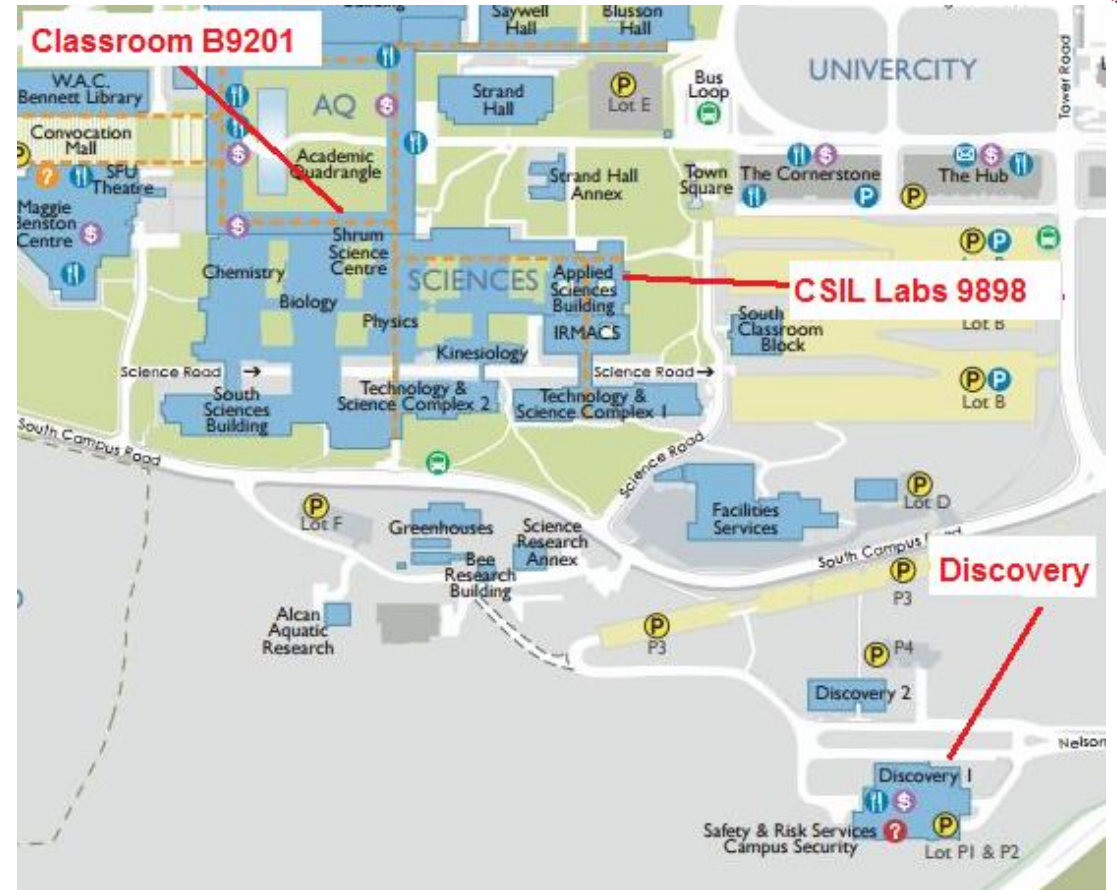
- Course Outline
- Exam Schedule
- Python Info
- CourSys/Canvas link
- Learning Outcomes
- Office Hours
- Textbook links
- and more...
- Grading Scheme
- Lab/Tutorial Info
- Assignments

- **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. Continue with Algorithms, Flowcharts
2. Pseudocodes

Today's Topics

1

Continue with Algorithms, Flowcharts

Algorithm: Find the Smallest of Three Numbers

Step 1: Start

Step 2: Declare variables $n1$, $n2$, and $n3$.

Step 3: Read variables $n1$, $n2$, and $n3$.

Step 4: **if** $n1 < n2$ then:

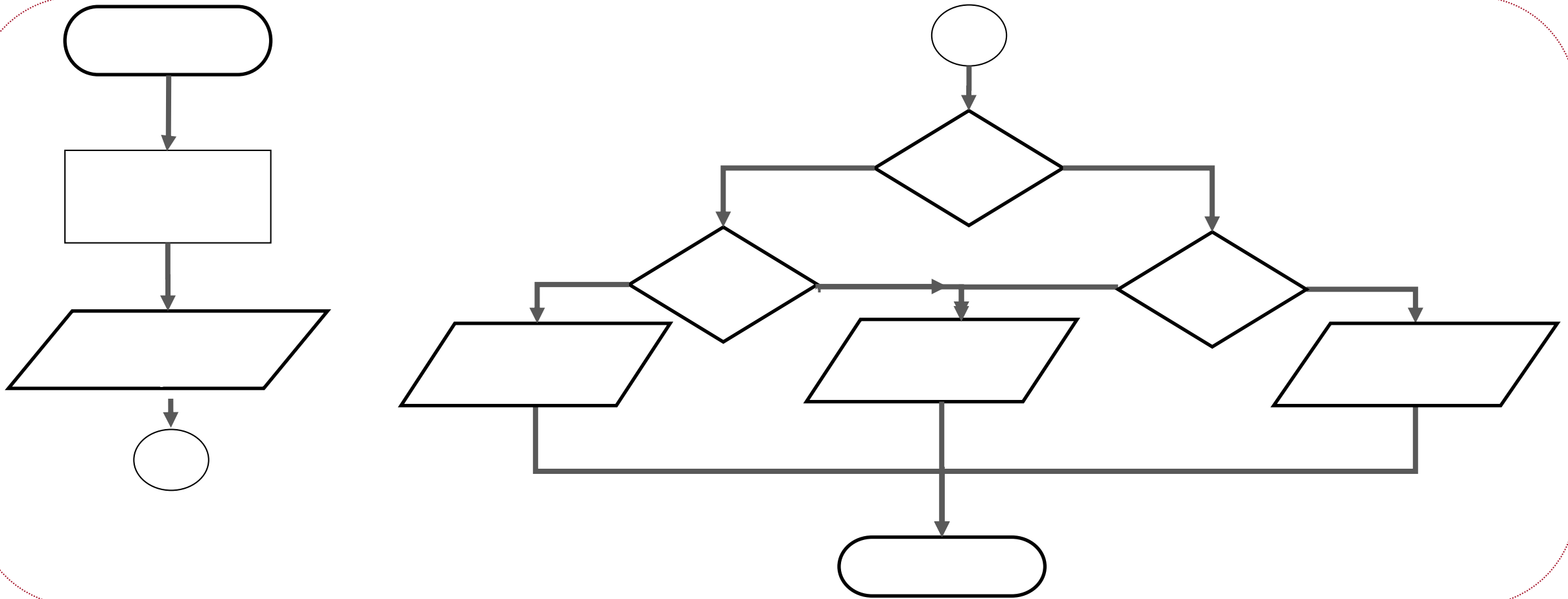
Step 5: **if** $n1 < n3$ then print $n1$ **else** print $n3$.

Step 6: **else**

Step 7: **if** $n2 < n3$ then print $n2$ **else** print $n3$.

Step 8: End

Flowchart: Smaller of Three Numbers (Solution)



Today's Topics

2

Pseudocodes

Pseudocodes



Pseudocodes - 2

-
-
- So, pseudocode is a **way to describe** the steps in an algorithm using some short and simple English (natural) language terms. (**Pseudo** is “almost”.)
- It describes an algorithm in specific enough detail to be easily implemented in any language.
- **Actually, some of the algorithms we wrote in the previous two classes equally qualify as pseudocodes.**

Pseudocodes: Features

- We typically use short phrases or keywords to describe steps in a pseudocode.
- For example:
- **READ, WRITE, SET, IF, ELSE, ENDIF, WHILE, ENDWHILE, REPEAT, UNTIL**
- Pseudocodes omit language specific syntax.
- It enables the programmers to concentrate on writing the coding.

Algorithm/Pseudocode: Smaller of Three Numbers

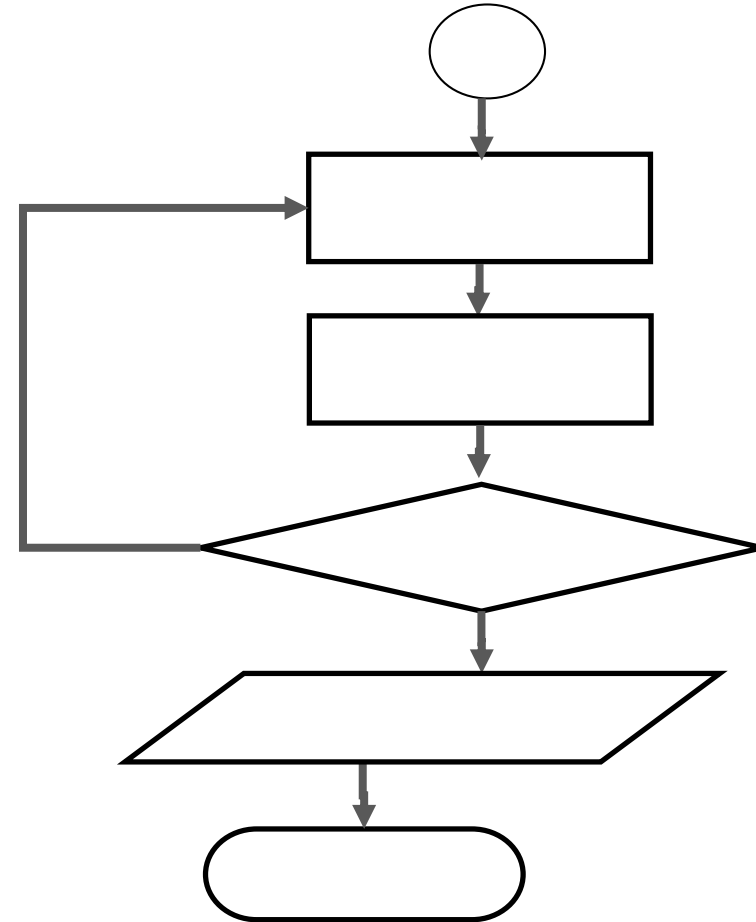
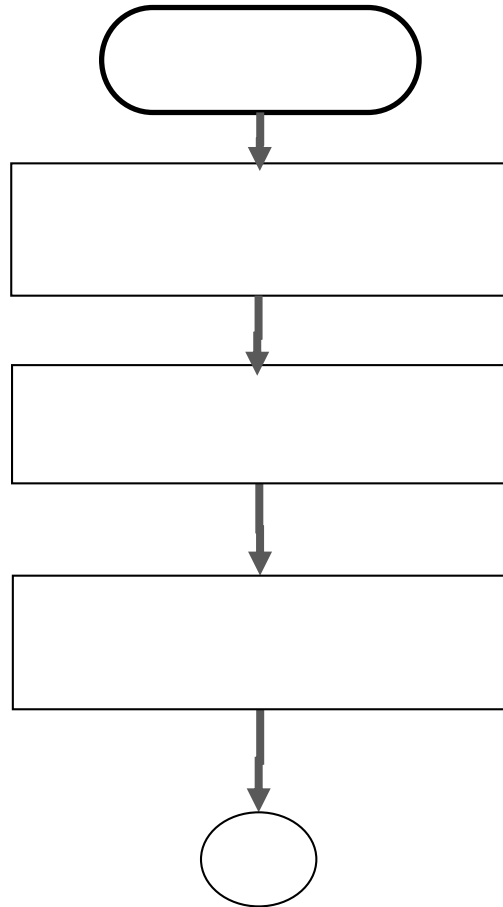
- 1: Start**
- 2: Declare variables $n1$, $n2$, and $n3$
- 3: Read variables $n1$, $n2$, and $n3$
- 4: If $n1$ is smaller than $n2$ and $n3$, then $n1$ smaller.
- 5: If $n2$ is smaller than $n1$ and $n3$, then $n2$ smaller.
- 6: If $n3$ is smaller than $n1$ and $n2$, then $n3$ smaller.
- 7: End**

Pseudocode: Find Sum of First 100 Natural Numbers

Step 1: Start

Step 9: End

Flowchart: Find Sum of First 100 Natural Numbers



Algorithm: Convert Height In Meters To Feet and Inches

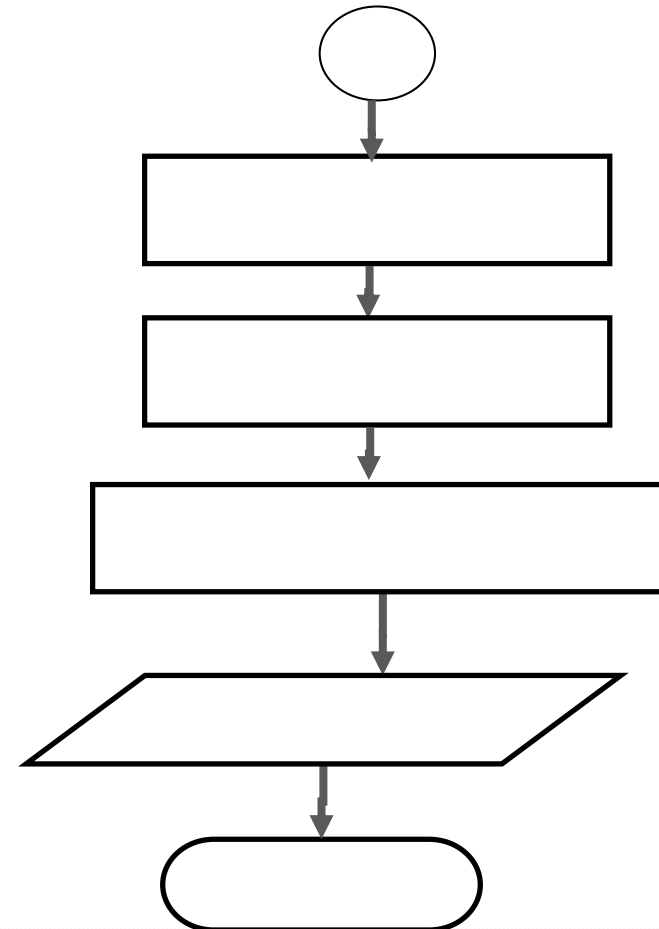
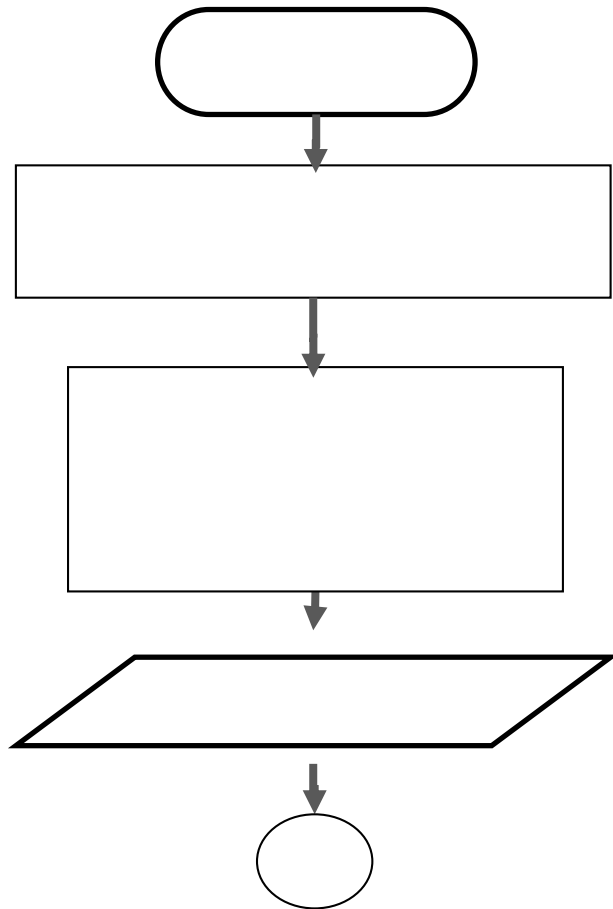
1: Start

2:

(floor)

9: End

Flowchart: Convert Height In Meters To Feet and Inches



Why Pseudocodes?

Writing code to solve a problem would have two parts:

1. Identifying **what** to : Writing **Pseudocode** (Algorithm)
 2. Knowing **how** to do : Writing **Python** code
- So, write an algorithm, express it in pseudocode before you start coding.
 - Especially as you're **starting to program**, you **don't want to be worrying** about **what you're trying to say** and **how to say it at the same time.**



Questions?