SIMON FRASER UNIVERSITY
ENGAGING THE WORLD

CMPT 120: Introduction to Computing Science and Programming 1

Algorithms, Flowcharts and Pseudocode

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.

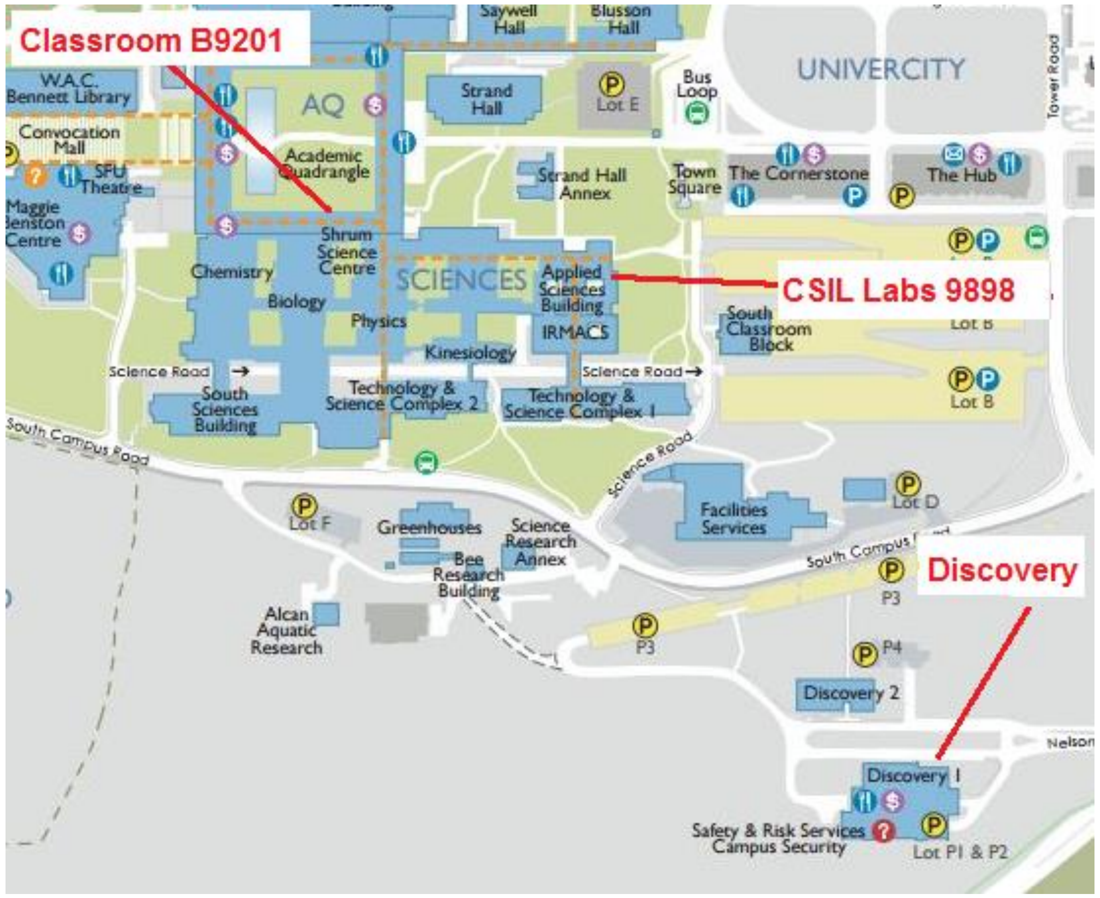
  https://canvas.sfu.ca/courses/39187

- **CourSys**: For assignments submission, and 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

- There are several online references that are **as important as the texts**. (Links provided on the course web site.)
- These resources are **very important to your success** in this course. 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.

Today’s Topics

1. Continue with Algorithms
2. Flowchart

https://etherpad.canvas.sfu.ca/p/i-8z1KelGBGco3wHfCPSJrPyv8VoMoIMe2laPnvFKp

Today’s Topics

1

Continue with Algorithms

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 is the smallest number.
Step 6:          else print n3 is the smallest number.
Step 7:      else
Step 5:          If n2 < n3 then print n2 is the smallest number.
Step 6:          else print n3 is the smallest number.
Step 9:  End
```

Let's Write Another Algorithm: Even or Odd Number

Write an algorithm to print whether the user entered an even or an odd number.

```
Step 1: Start
Step 2:     Declare variables n and r.
Step 3:     Read the value of variable n.
Step 4:     Compute integer remainder of n divided by 2 and store it
            in r.
Step 4:     If r = 0 then print n is an even number.
Step 5:     else print n is an odd number.
Step 6: End
```

Today’s Topics

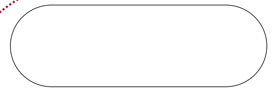
2

Flowcharts

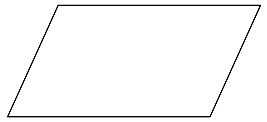
Flowcharts

- Flowchart is a graphical representation of an algorithm.
  - Flowchart is same as algorithm, except that in flowcharts we show the steps of an algorithm using geometric shapes like circles, rectangle, lines, diamonds etc.

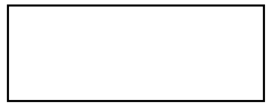
Flowcharts: Geometric Shapes and Their Meanings



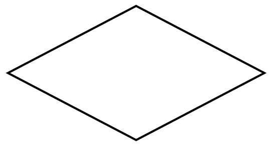
1. **Terminal**: To mark **Start** or **End** a flowchart.



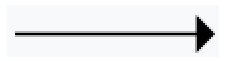
2. **I/O**: To show an **Input** or **Output** operation:
   - Read data from keyboard/user, or print/display on screen.



3. To show a **Process**:
   - Compute average, computer salary, add numbers.



4. To show a **Decision** point, or alternatives:
   - If marks > 50, **then** "Pass", **Else** "Fail".

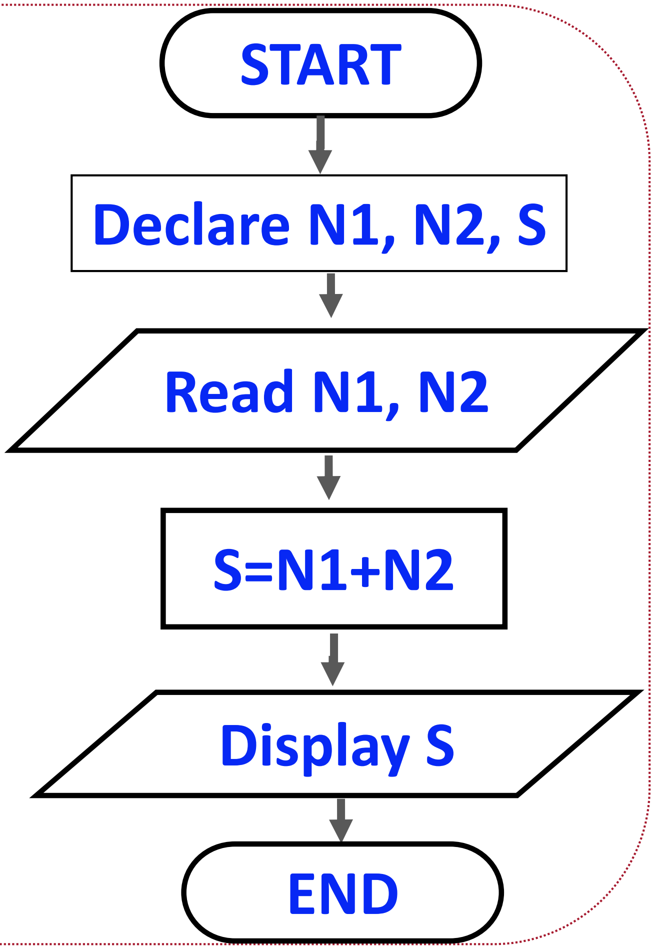


5. Flowline: To connect two steps / shapes / processes.

Note: See textbook/online resources for more symbols.

Draw a Flowchart for the Adding Two Numbers Algorithm

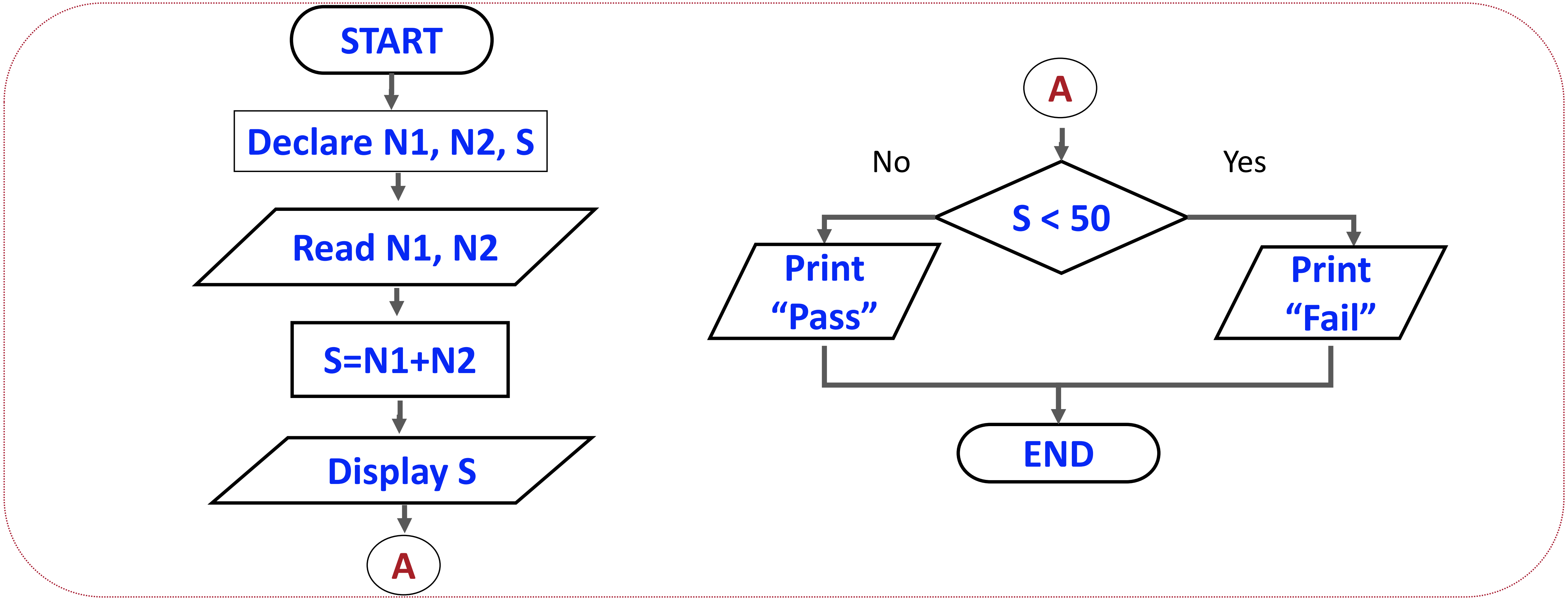
Step 1: Start
Step 2: Declare a variable N1.
Step 3: Declare a variable N2.
Step 4: Declare a variable S to store the sum.
Step 5: Get the value of N1 from the user.
Step 6: Get the value of N2 from the user.
Step 7: Add N1 and N2 and assign the result to S.
Step 8: Display the sum S.
Step 9: End



Modify Algorithm: Add, If Sum < 50 Then Fail Else Pass

```
Step 1: Start
Step 2:  Declare a variable N1.
Step 3:  Declare a variable N2.
Step 4:  Declare a variable S to store the sum.
Step 5:  Get the value of N1 from the user.
Step 6:  Get the value of N2 from the user.
Step 7:  Add N1 and N2 and assign the result to S.
Step 8:  Display the sum S.
Step 9:  If S < 50 then display "Fail"
          Else display "Pass"
Step 10: End
```

Modify the Flowchart



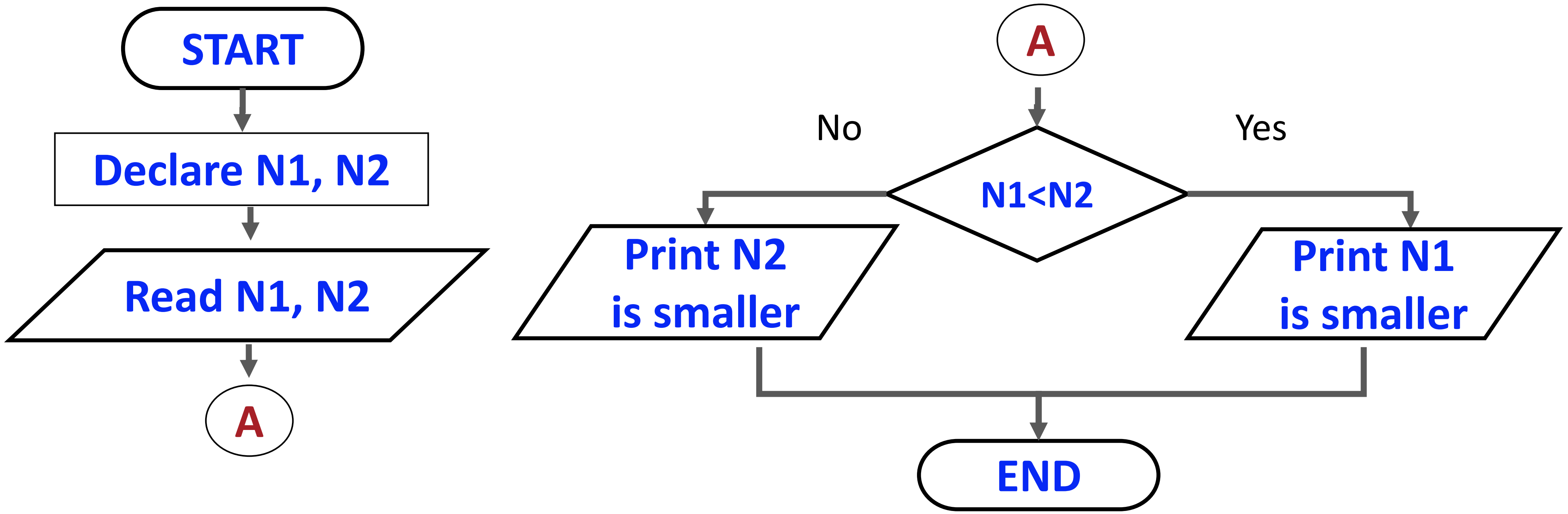
Note: circle / oval shape is a same page connector.

Draw Flowchart: Find the Smaller of Two Numbers Algorithm

Write an algorithm to find the smaller of two numbers entered by a user.

```
Step 1:  Start
Step 2:      Declare a variable num1 to store the first number.
Step 3:      Declare a variable num2 to store the second number.
Step 4:      Get the value of num1 from the user.
Step 5:      Get the value of num2 from the user.
Step 6:      If num1 < num2 then print num1 is smaller.
Step 7:      If num2 < num1 then print num2 is smaller.
Step 8:      If num1 = num2 then print “Both the numbers are equal.”
Step 9:  End
```

Flowchart: Smaller of Two Numbers



Draw Flowchart: Home Work (Solution In Next Class)

```
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 is the smallest number.
Step 6:           else print n3 is the smallest number.
Step 7:       else
Step 5:           If n2 < n3 then print n2 is the smallest number.
Step 6:           else print n3 is the smallest number.
Step 9:  End
```

Questions?

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