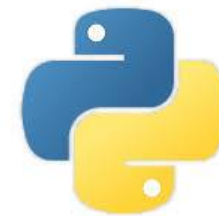


# CMPT 120: Introduction to Computing Science and Programming 1

## 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.

# Reminders

Liaqat Ali, Summer 2018.

# 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](http://www.coursys.sfu.ca)

# How to Learn in This Course?



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

# Additional / Online References

- **Additional references** are as important as the texts, and very important to your success.
  - They aren't meant to be read from beginning to end like the readings in the textbook.
- Use them to get an overall picture of the topic and as references as you do the assignments.

# 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. Programs Recap
2. Expressions
3. Operands
4. Operators
  - i. Arithmetic Operators (+, -, \*, /)
  - ii. Comparison operators
5. Reflection

1

# Programs Recap

Liaqat Ali, Summer 2018.



# Program Recap: Compute Grade (Solution)

```
midterm = 0
final   = 0

midterm = input("Enter midterm:")
final   = input("Enter final:")

total = float(midterm) + float(final)

if total >= 95: print("A+")
elif total >= 90 and total < 95: print("A")
elif total >= 85 and total < 90: print("A-")
elif total >= 80 and total < 85: print("B+")
elif total >= 75 and total < 80: print("B")
elif total >= 70 and total < 75: print("B-")
elif total >= 65 and total < 70: print("C+")
elif total >= 60 and total < 65: print("C")
elif total >= 55 and total < 60: print("C-")
elif total >= 50 and total < 55: print("D")
else: print("F")
```

# Program Recap: Sum of Natural Numbers (Solution)

```
sum = 0
```

```
n = 1
```

```
while ( n <= 100 ):
```

```
    sum=sum+n
```

```
    n=n+1
```

```
print(sum)
```

```
sum = 0
```

```
n = 1
```

```
sum=sum+n
```

```
n=n+1
```

```
print(sum)
```

2

# Expressions

Liaqat Ali, Summer 2018.

# Expressions

- We are now familiar with, and have used in our programs:
  - a. **values**, such as 5, 7 , or 100
  - b. **variables**, such midterm, final, or total
  - c. **operators**, such +, /, or %
- An **expression** is a
  - 
  - 
  - We can categorize expressions based on their result types:
    - 1.
    - 2.

# Arithmetic Expressions

- When result of an expression is a numeric value, we can call it an **Arithmetic Expression**.
  - For example, \_\_\_\_\_ is an arithmetic expression if  $n$  is numeric.
    - Suppose  $n$  is **5**, then the value of the arithmetic expression  $n + 1$  would be **6**, which is a numeric value.
  - \_\_\_\_\_ is an arithmetic expression if  $meters$  is numeric.
    - Suppose  $meters$  is **2**, then the value of the arithmetic expression  $meters * 39.37$  would be **78.74**, which is again a numeric value.
  - A numeric value can be an integer (whole number), or
  - A floating point number (with decimal point).

# Boolean Expressions

- When the result of an expression is either **True** or **False**, we call it a **Boolean Expression**. For example:

Meaning

Outcome

- $5 < 7$
- $\text{marks} > 95$

3

# Operands

In our previous class, we talked about operators.

- **Operators** are special symbols that represent **computations**.
- **Arithmetic Operators** are symbols we use to represent arithmetic operations. For example,  $+$ ,  $-$ ,  $*$ , or  $/$ .
- We'll continue with operators today and will talk about **Comparison Operators**.
- But first, let's get familiar with a new term **Operands**.



# Operands

- **Operands** are the values that appear \_\_\_\_\_.
- For example, in an arithmetic expression \_\_\_\_\_, the values **50** and **10** are operands.
- In \_\_\_\_\_, the values **70** and **15** are operands.
- They are the data to be **operated on** by the operator.
- So, think of operands **just another name** for the values operators use.
- **Operands** can be values or variable names.
  - For example, in \_\_\_\_\_, both the operands **mid** and **final** are variables.

4

# Comparison Operators

# Operators: Compute Grade

```
midterm = 0
```

```
final = 0
```

```
midterm = input("Enter midterm:")
```

```
final = input("Enter final:")
```

```
total = float(midterm) + float(final)
```

```
if total >= 95: print("A+")
```

```
elif total >= 90 and total < 95: print("A")
```

Special symbol.  
Arithmetic Op.

```
elif total >= 85 and total < 90: print("A-")
```

```
elif total >= 80 and total < 85: print("B+")
```

```
elif total >= 75 and total < 80: print("B")
```

```
elif total >= 70 and total < 75: print("B-")
```

```
elif total >= 65 and total < 70: print("C+")
```

```
elif total >= 60 and total < 65: print("C")
```

```
elif total >= 55 and total < 60: print("C-")
```

```
elif total >= 50 and total < 55: print("D")
```

```
else: print("F")
```

# Comparison Operators

- The symbols \_\_\_\_\_ are called **comparison operators**. (They are **6** in number.)
- **Comparison operators** are used to **compare** values or operands.
  - For example in a Boolean expression:
    - **5 < 6**: the symbol **<** is a comparison operator, and **5** and **6** are values.
    - **total >= 90**, **>=** is a comparison operator, and **total** and **90** are values.
- A comparison either returns a **True** or **False** result.
  - An expression that results into a true or false value is called a Boolean Expression.

# Comparison Operators: Descriptions and Examples

Suppose:  $a = 5$   $b = 7$

			Meaning
1.	$<$ less than	$( a < b )$	is a less than b?
2.	$<=$ less than or equal	$( a <= b )$	is a less than or equal to b?
3.	$>$ greater than	$( a > b )$	is a greater than b?
4.	$>=$ greater than or equal	$( a >= b )$	is a greater than or equal b?
5.	$==$ equal	$( a == b )$	is a equal to b?
6.	$!=$ not equal	$( a != b )$	is a not equal to b?
		$( a <> b )$	

# Grade Program Example

- Recall this program we wrote last week.
- In the comparison expression **if total >= 50:**
  - What conditional operator did it use?
    - >= (greater than or equal)
  - What are the operands?
    -
  - What are the possible outcomes?
    - ---

      - We call this outcome as True
    - ---

      - We call this outcome as False

```
midterm = input()
final = input()
total = float(mid) +
float(final)

if total >= 50:
    print("Pass")

else:
    print("F")
```

# Grade Computation Program

Greater than or equal comparison operator

```
midterm = 0
final = 0
total = 0
midterm = input()
final = input()
total = float(midterm) +
        float(final)
if total >= 95: print("A+")
```

```
elif total >= 90 and total < 95: print("A")
elif total >= 85 and total < 90: print("A-")
elif total >= 80 and total < 85: print("B+")
elif total >= 75 and total < 80: print("B")
elif total >= 70 and total < 75: print("B-")
elif total >= 65 and total < 70: print("C+")
elif total >= 60 and total < 65: print("C")
elif total >= 55 and total < 60: print("C-")
elif total >= 50 and total < 55: print("D")
else: print("F")
```

We'll talk next.

Less than comparison operator.

5

# Reflection

Liaqat Ali, Summer 2018.



# Compute Grade – Version 2

```
midterm = 0
final = 0
grade = ""

midterm = input("Enter midterm:")
final = input("Enter final:")

total = float(midterm) + float(final)

if total >= 95: grade = "A+"
elif total >= 90 and total < 95: grade = "A"
elif total >= 85 and total < 90: grade = "A-"
```

```
elif total >= 80 and total < 85: grade = "B+"
elif total >= 75 and total < 80: grade = "B"
elif total >= 70 and total < 75: grade = "B-"
elif total >= 65 and total < 70: grade = "C+"
elif total >= 60 and total < 65: grade = "C"
elif total >= 55 and total < 60: grade = "C-"
elif total >= 50 and total < 55: grade = "D"
else: grade = "F"
print(grade)
print(total, grade)
print("Total marks = ", total, "Grade = ", grade)
```

# Class Participation Activity

1. Copy and run this program.
2. Reflect on:
  - a. How this program differs from the program on slide 8.
  - b. Why it produces the same output as the program on slide 8.
  - c. Which approach you think is better: direct printing, or using a variable? Think of one reason?
  - d. What if you replace the word **and** with **or** in the if conditionals?
    - Modify and run the program for the pair of values 34,40; 60,30; 23, 4; and 45, 46.
3. Post your reflections on the **Canvas Discussions** forum after the class today.



# Questions?