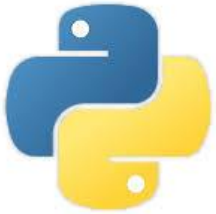
SFU SIMON FRASER UNIVERSITY
ENGAGING THE WORLD

# CMPT 120: Introduction to Computing Science and Programming 1

## Functions



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

# 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

# Course Topics

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

# Today's Topics

1. Turtle Graphics: Drawing and Animation
2. Introduction to Functions: User-defined
3. Defining and Calling a Void Function
4. Designing a Program to Use Functions
5. Passing Arguments to Functions
6. Case Study: Developing Software Using Functions

1

# Designing a Program to Use Functions

Liaqat Ali, Summer 2018.

# Designing a Program to Use Functions

- In a flowchart, function call shown as rectangle with vertical bars at each side
  - Function name written in the symbol.
  - Typically draw separate flow chart for each function in the program
    - End terminal symbol usually reads `Return`.
- Top-down design: technique for breaking algorithm into functions

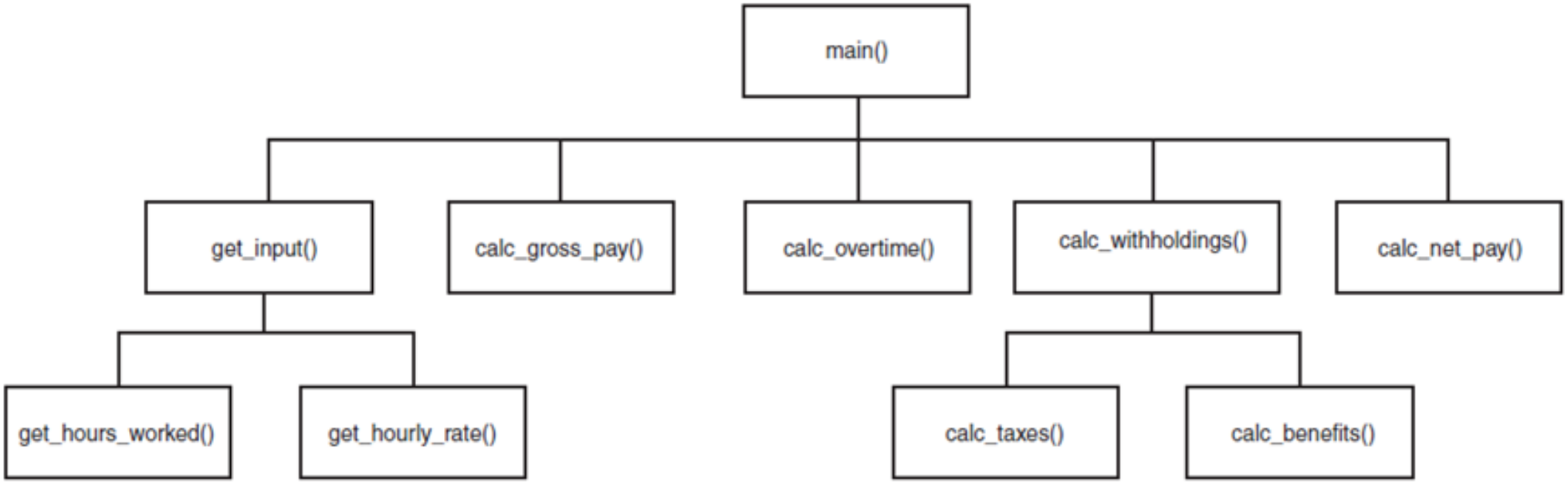
 Copyright © 2018 Pearson Education, Inc.

# Designing a Program to Use Functions (cont'd.)

- Hierarchy chart: depicts relationship between functions
  - AKA structure chart
  - Box for each function in the program, Lines connecting boxes illustrate the functions called by each function
  - Does not show steps taken inside a function
- Use `input` function to have program wait for user to press enter.

 Copyright © 2018 Pearson Education, Inc.

# Designing a Program to Use Functions (cont'd.)




 Copyright © 2018 Pearson Education, Inc.

# Local Variables

- **Local variable**: variable that is assigned a value inside a function
  - Belongs to the function in which it was created.
    - Only statements inside that function can access it, error will occur if another function tries to access the variable.
- **Scope**: the part of a program in which a variable may be accessed
  - For local variable: function in which created.

 Copyright © 2018 Pearson Education, Inc.

# Local Variables (cont'd.)

- Local variable cannot be accessed by statements inside its function which precede its creation.
- Different functions may have local variables with the same name
  - Each function does not see the other function's local variables, so no confusion.

 Copyright © 2018 Pearson Education, Inc.

# Passing Arguments to Functions

- **Argument**: piece of data that is sent into a function.
  - Function can use argument in calculations.
  - When calling the function, the argument is placed in parentheses following the function name.

 Copyright © 2018 Pearson Education, Inc.

# Passing Arguments to Functions (cont'd.)

```
def main():
    value = 5
    show_double(value)
```

```
def show_double(number):
    result = number * 2
    print(result)
```

 Copyright © 2018 Pearson Education, Inc.

# Passing Arguments to Functions (cont'd.)

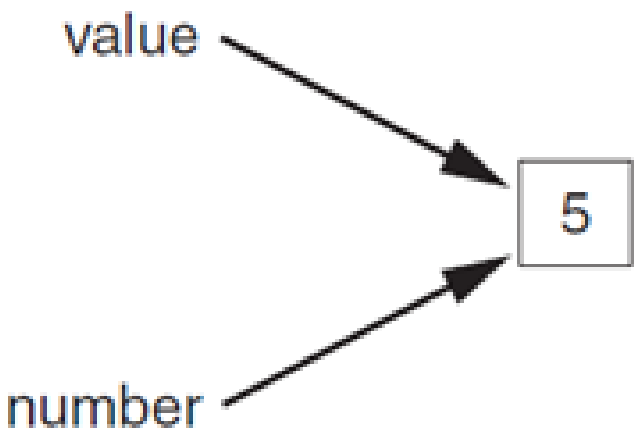
- Parameter variable: variable that is assigned the value of an argument when the function is called.
  - The parameter and the argument reference the same value
  - General format:
  - `def function_name(parameter):`
  - **Scope of a parameter**: The function in which the parameter is used.

 Copyright © 2018 Pearson Education, Inc.

# Passing Arguments to Functions (cont'd.)

```
def main():
    value = 5
    show_double(value)


def show_double(number):
    result = number * 2
    print(result)
```




 Copyright © 2018 Pearson Education, Inc.

# Questions?

Copyright © 2018 by Liaqat Ali