

Programming

CPSC 231: Introduction to Computer Science for Computer Science Majors I Fall 2021

Jonathan Hudson, Ph.D.
Instructor
Department of Computer Science
University of Calgary

Tuesday, 24 August 2021

Copyright © 2021



How Do We Solve Problems with a Computer?

First question: How do we learn?

**What does it mean to
understand
something?**

Bloom's Taxonomy

create

Produce new or original work

Design, assemble, construct, conjecture, develop, formulate, author, investigate

evaluate

Justify a stand or decision

appraise, argue, defend, judge, select, support, value, critique, weigh

analyze

Draw connections among ideas

differentiate, organize, relate, compare, contrast, distinguish, examine, experiment, question, test

apply

Use information in new situations

execute, implement, solve, use, demonstrate, interpret, operate, schedule, sketch

understand

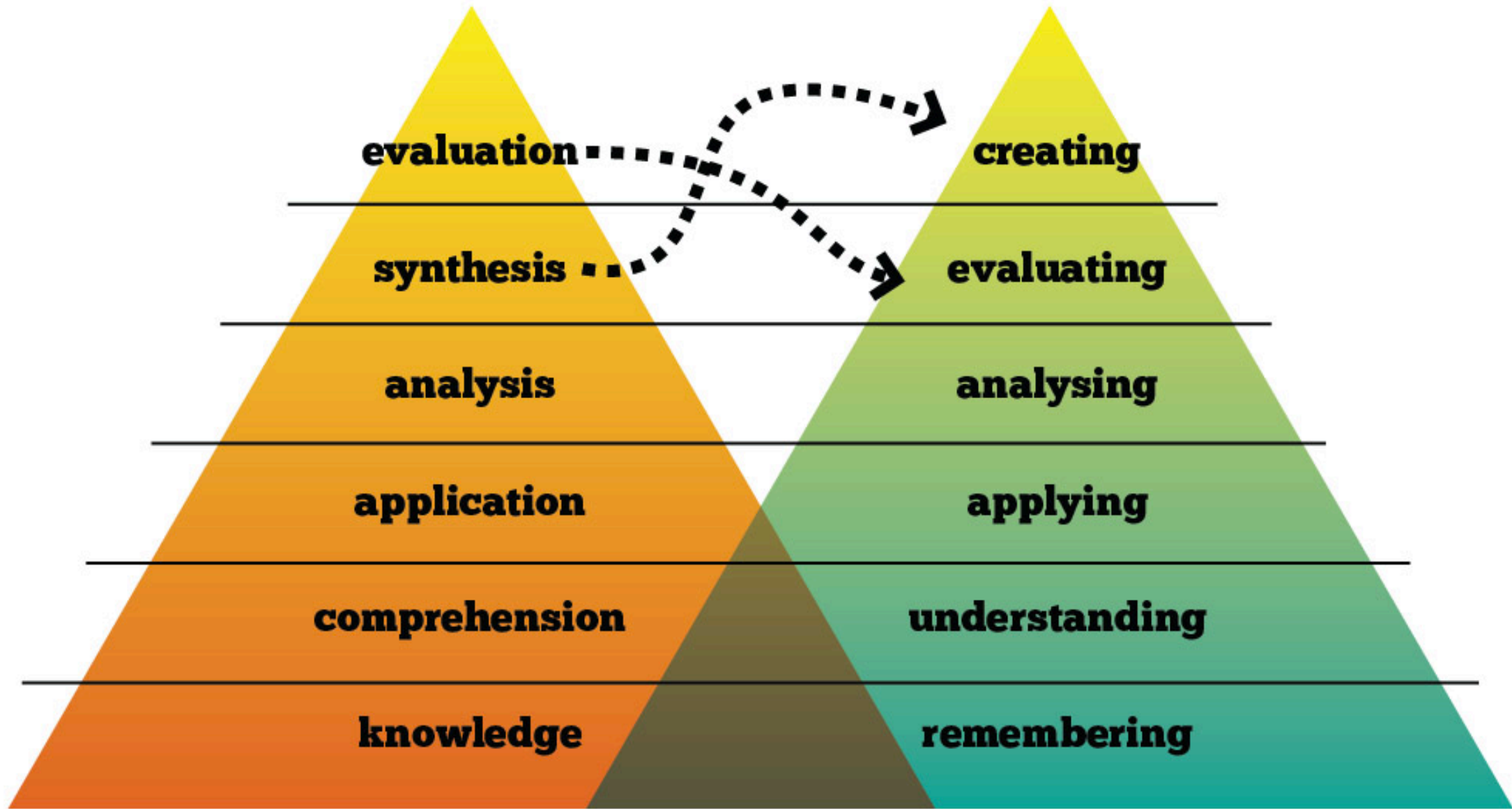
Explain ideas or concepts

classify, describe, discuss, explain, identify, locate, recognize, report, select, translate

remember

Recall facts and basic concepts

define, duplicate, list, memorize, repeat, state



Schultz 2005

The Old Version

The New Version

Solving Problems

How do we solve problems?

Solving Problems



How do we solve problems?



Break original problem into smaller, more easily solvable parts and repeat on the smaller sub-problems.



Similar process used in expository writing (break topic up into easily-understandable bits, express in coherent way).

Top Down Design

01

Start with the entire problem

02

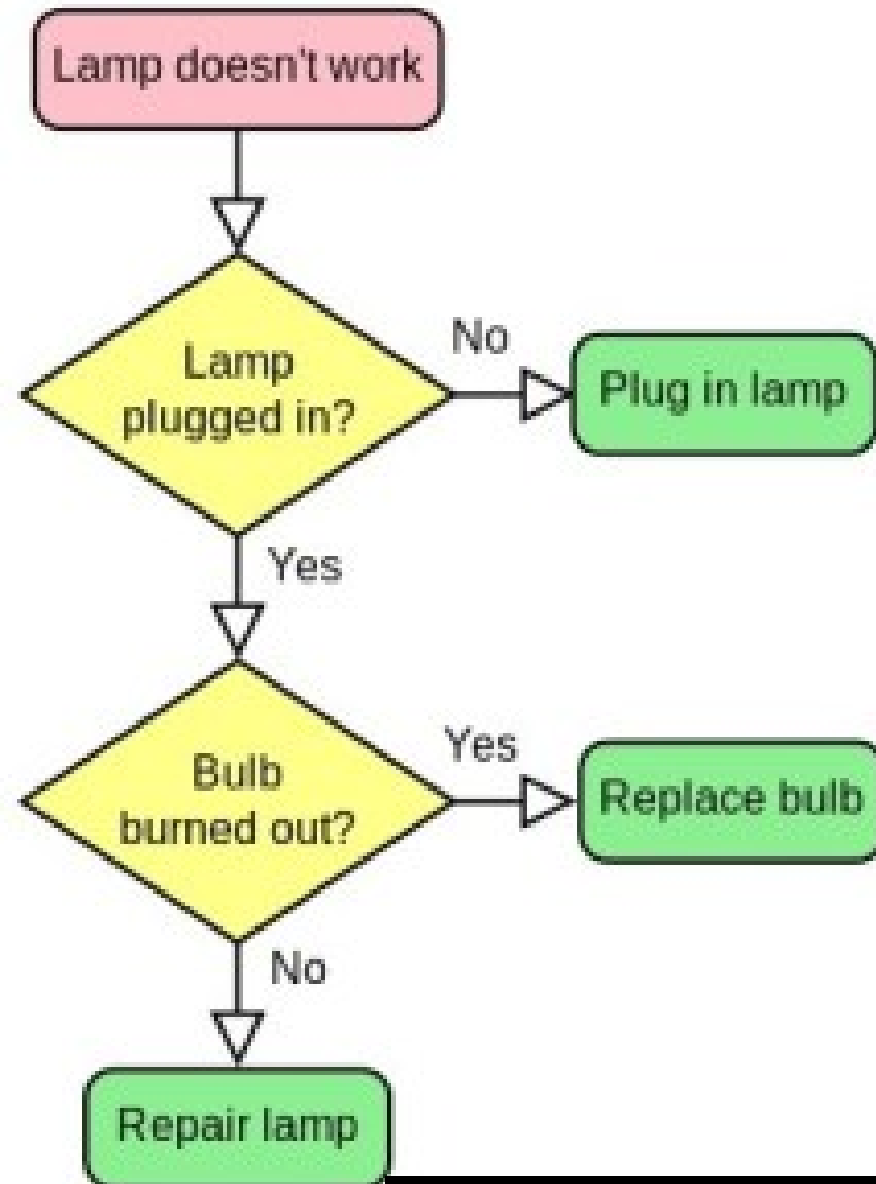
Break the problem into approximately 3 to 5 smaller steps

03

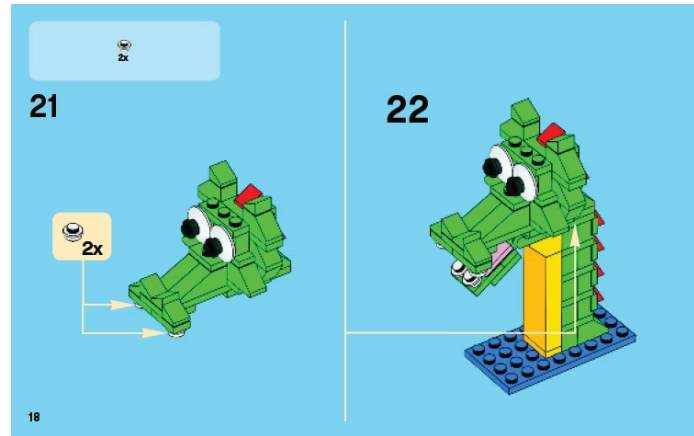
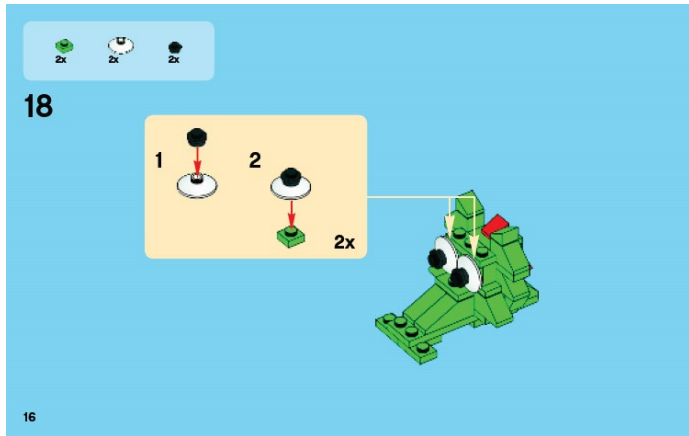
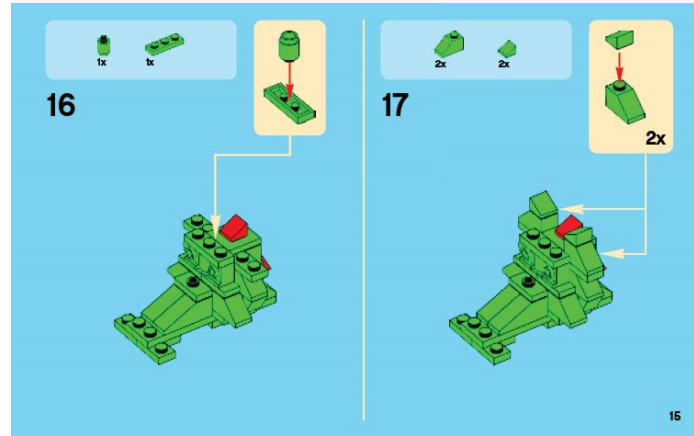
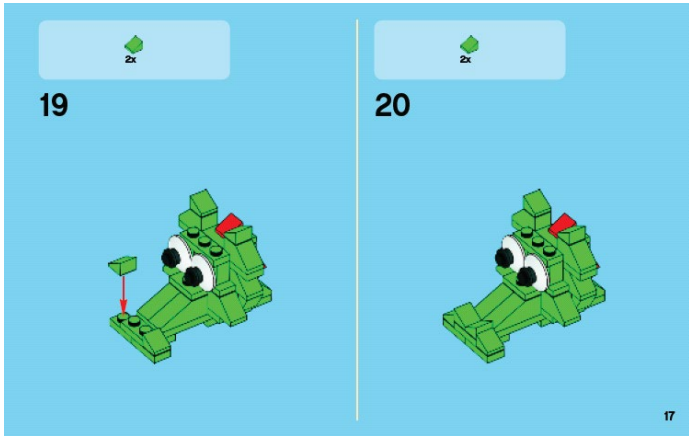
Repeat the process for each step that is still too complex

What is an Algorithm?

What is an Algorithm?



This Photo by Unknown Author is licensed under [CC BY-SA-NC](https://creativecommons.org/licenses/by-sa/4.0/)



What is an Algorithm?

- **Algorithm: A finite sequence of effective (unambiguous, possible) steps to solve a problem.**
- **Expressed in English, human-oriented form**
- **Result of top-down design (or other problem solving strategy)**
- **A well written algorithm can be written in any computer programming language**

What is Programming?



What is Programming?

- **Programming:** the process of creating software by translating algorithms into a computer language.
- **Algorithm:** human readable form, layout/syntax is free as long as a reasonable person can understand it
- **Computer Program:** computer readable form, precise syntax that must be followed exactly, will do exactly what you say (not what you meant!)

Where Are We Going?

- Computers are tools that we use to solve problems
 - Need to understand the problem that we want to solve
 - Need to understand how a computer works to model the problem
 - Need to learn how to program the computer to solve the problem

Programming Languages

Many programming languages available

- Offer different features
- **Each has its own strengths and weaknesses**

Common features

- Allow us to control the behaviour of a computer
- **Defined syntactic and semantic rules**
 - **Syntactic** – what does a valid statement look like
 - **Semantic** – what is meaning of a statement

Example: Syntax versus Semantics

Python 2:

$$5 / 2 = 2$$

$$5 / 2.0 = 2.5$$

$$5.0 / 2 = 2.5$$

Python 3

$$5 / 2 = 2.5$$

$$5 / 2.0 = 2.5$$

$$5.0 / 2 = 2.5$$

Syntax is the same.

Division is done via /

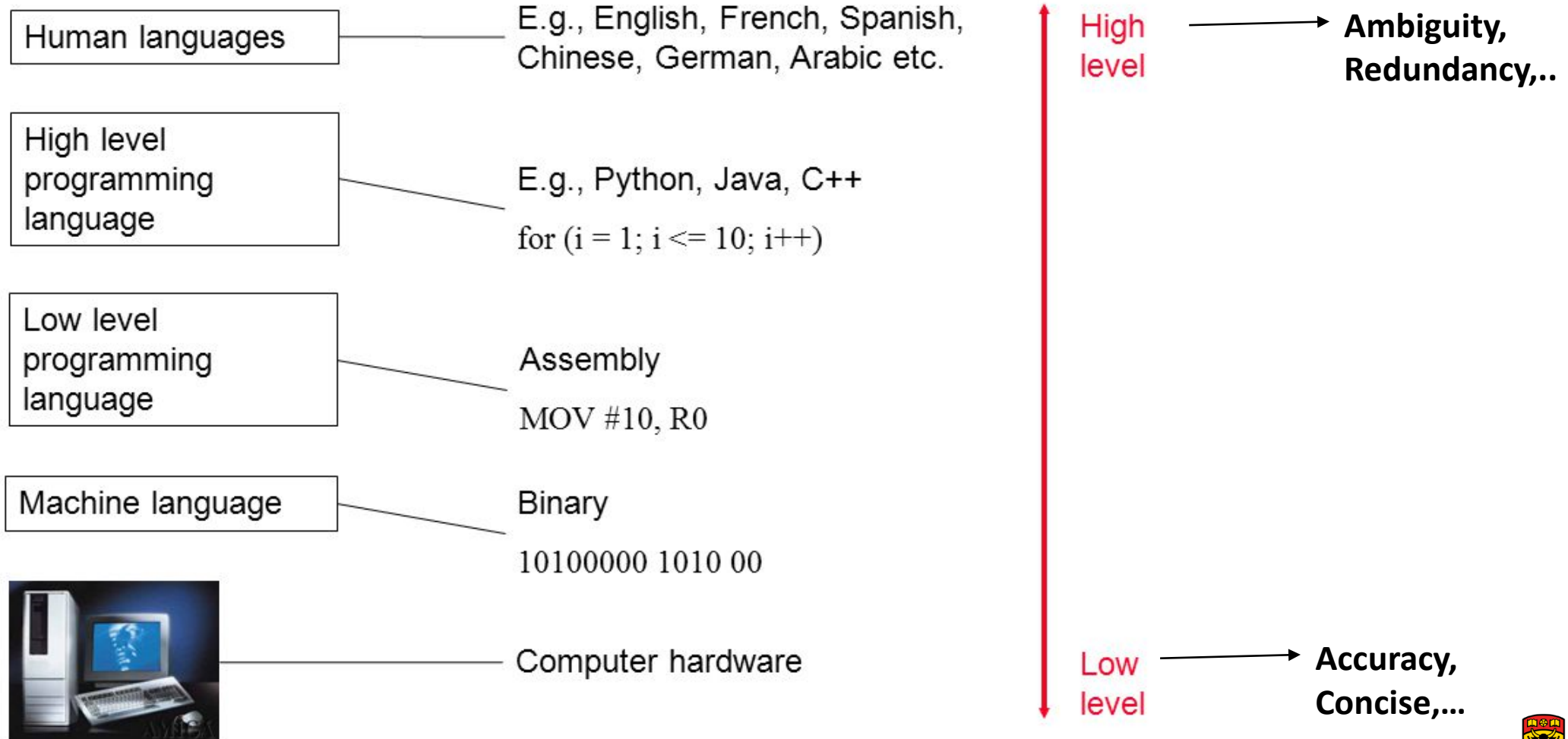
Semantics are different.

Python 2 division result is based on whether one input has decimals

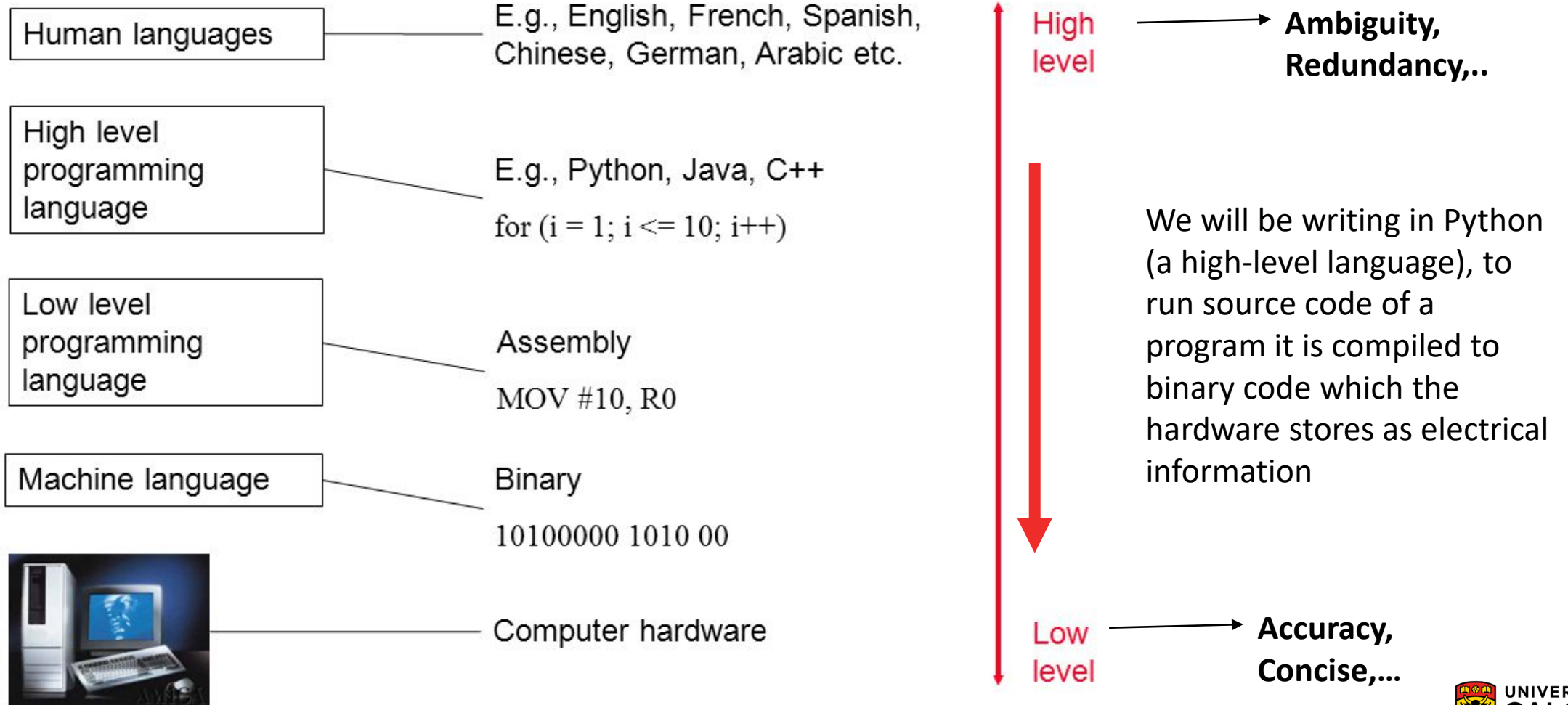
One decimal value number creates decimals out

Python 3 always produces decimal numbers

High Level Vs. Low Level Languages



High Level Vs. Low Level Languages



Using Python

- **Python 3 is the official programming language for this course.**
- There are different ways to tell Python to execute your code:
 - Interactive coding
 - System command lines and files
- This course does not encourage the use of IDE such as Eclipse
- We encourage direct interaction with the computer systems

Compiler/Interpreter

Programming

- Uncompiled computer program code is stored in **source code files**
 - Human readable / editable
 - Can also be understood by a computer
 - typically have the extension **.py**
- Once the file is created, it is run using the installed python program
 - **python myfile.py**

Compilers Vs. Interpreter

- A **compiler**:

- Is like **translating an entire book** and give it to a reader.
- A compiler reads the program and translates it completely before the program starts running
- Fully compiled code is machine code (instructions to CPU in binary), not easy for human to understand



- An **interpreter**

- is **like translating a line at a time** and give the line to the uses until the book is finished.
- It processes the program a little at a time, alternately reading lines and performing computations.
- Generates machine code as it processes line by line



Compiler	Interpreter
Compiler Takes Entire program as input	Interpreter Takes Single instruction as input .
Intermediate Object Code is Generated	No Intermediate Object Code is Generated
Conditional Control Statements are Executes faster	Conditional Control Statements are Executes slower
Memory Requirement : More (Since Object Code is Generated)	Memory Requirement is Less
Errors are displayed after entire program is checked	Errors are displayed for every instruction interpreted (if any)

Difference between Compiler and Interpreter

Compiler or Interpreter

- Many modern languages use both processes.
- Python uses both processes.
- Python is a **compiled interpreted language**

- Python is first compiled into a lower-level language, called **byte code**, and then interpreted by a program called a **virtual machine**.
 - (**byte code is higher level than machine code**, we can still move it between machine types Ex. Win/MacOS. Machine code can only move within same type.)
 - A **virtual machine** is created for each machine type.

- There is also an **interpreter-only** mode for Python where we can type instructions line by line. However, it is rare to use this outside of teaching programming.

Coding

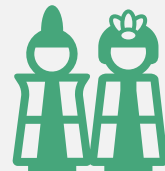
Command lines and files



You can check your version using
python -V



If the result is 2.X.X then



You will have to use: **python3
hello.py**

Interactive coding (Interpreter-only mode)

- `>>>` is a Python prompt indicating that python is ready to accept commands

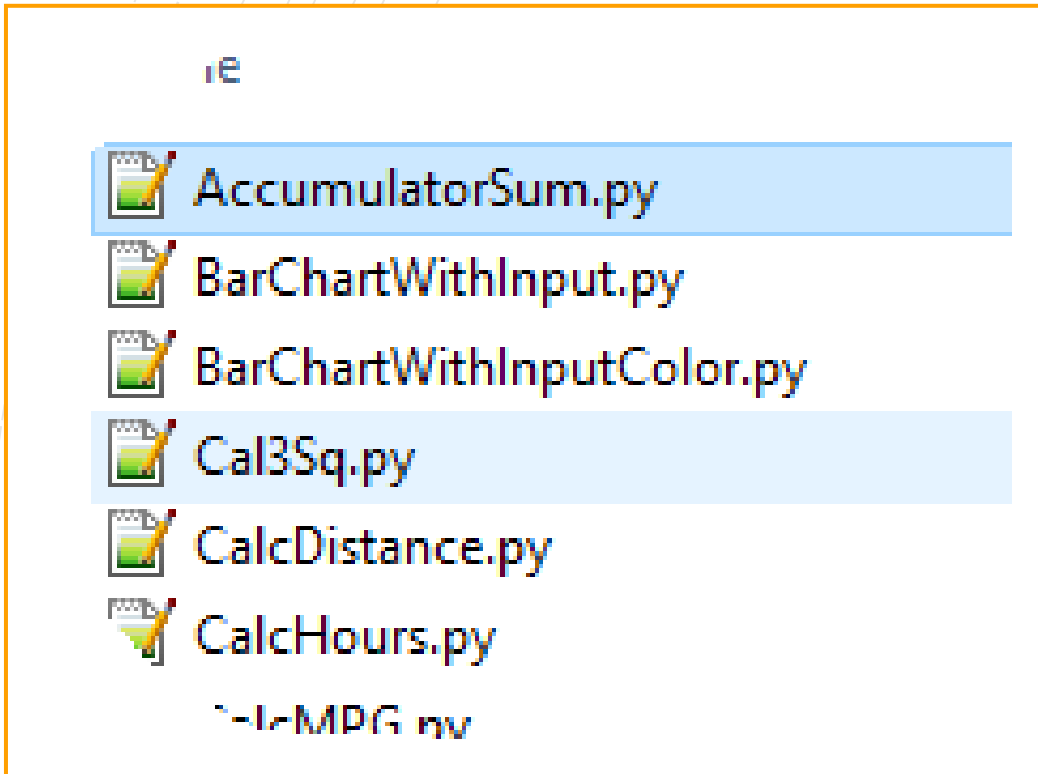
```
cmd - python
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Windows\System32>python
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:14:34) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

- Python Functions:
 - `print()` → prints text to the screen
 - `exit()` → exits python

```
>>> 1+1
2
>>> print("Hello World")
Hello World
>>>
```

Command Lines and Files



Source code is a file containing your code often referred to as a **program**.

- The filename ends with a **.py** suffix

e.g. **hello.py**

- To execute from terminal/shell:

python hello.py

- To save the output into a file:

python hello.py > output.txt

Onward to ... variables!

Jonathan Hudson
jwhudson@ucalgary.ca
<https://pages.cpsc.ucalgary.ca/~hudsonj/>

