COMP10001 Foundations of Computing
The Basics of Programming

Semester 2, 2021

Chris Leckie, Marion Zalk and Farah Khan

THE UNIVERSITY OF MELBOURNE

— VERSION: 1468, DATE: MARCH 11, 2019 —

Lecture Agenda

- Last lecture:
  - Computers speak binary, but we don't
  - High level programming languages make life easier
  - We will use Python inside Grok
- This lecture:
  - Programming basics with blockly

Lecture Outline

1. Announcements
2. Programming

... Emailing the Lecturer?

- If you email us asking a question that could equally have been asked on the forums, we will respond via the forum and **not** email ... not because we want to be rude, but just to clarify boundaries of what is Chris/Marion-emailable and what is not, and because we can't deal with 800+ students' worth of one-on-one email!
- Please don't be offended; understand the reasons behind it!
- Note that we equally can't service random meetings: grab us during consultation time, or get help from your tutor
- Remember - Please be respectful at all times in your communication with staff and other students

Lecture Outline

Programming

- Computer programs are simply sets of steps to complete some task
- Determining what the steps should be requires learning how computers "think"... and how a particular programming language expresses the way a computer thinks
- At its most basic level, a program is made up of a sequence of **statements** that are executed sequentially one after the other

A Simple Program for Visiting a Friend's House

- Head south on the Peninsula Freeway
- Take the Red Hill exit
- Keep taking the left turn at each intersection until you cross the bridge
- If you reach the beach, turn around and go back (you've gone too far)
- Park at the Red Hill Bakery

Basic Programming Building Blocks

- The basic building blocks of programming are:
  - **statements** (= single "commands" to the computer)
  - **sequence** (= linear sequence of statements)
  - **control** (= perform sequence of statements IF condition holds)
  - **loops** (= repeat sequences of statements)
  - **functions** (= blocks of code that can be run with different inputs)
  - **recursion** (= blocks of code that call themselves with different inputs)

Turtle Programming

- As an illustration of this, without getting bogged down in the details too much, let's play around with Turtle graphics, using the "blockly" programming language
- Basic commands:
  - advance forward/backward `N` units
  - turn left/right `N` degrees

Class Exercise

- Using just `move` and `turn` statements, build blockly code to draw an equilateral triangle with side length 100

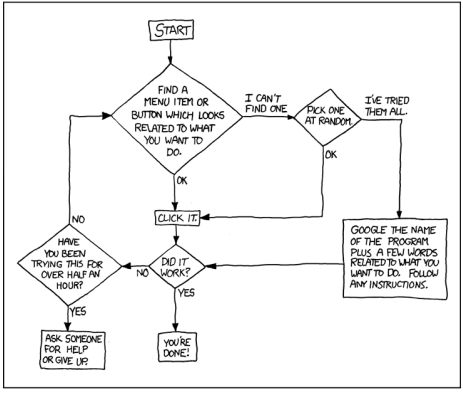
Program Design

- Many modern computing languages express similar concepts
- They allow "conditioning" on particular values, "looping" over sub-sets of steps, and "nesting" of loops
- Common ways to abstractly represent programs are:
  - flowcharts
  - "pseudo-code" (i.e. a computer program in an abstract language, without the "bookkeeping" that individual languages require)
    `http://www.bestrecipes.com.au/recipe/choc-chip-cookies-L4351.html`

Example Flowchart

DEAR VARIOUS PARENTS, GRANDPARENTS, CO-WORKERS, AND OTHER "NOT COMPUTER PEOPLE."

WE DON'T MAGICALLY KNOW HOW TO DO EVERYTHING IN EVERY PROGRAM. WHEN WE HELP YOU, WE'RE USUALLY JUST DOING THIS:



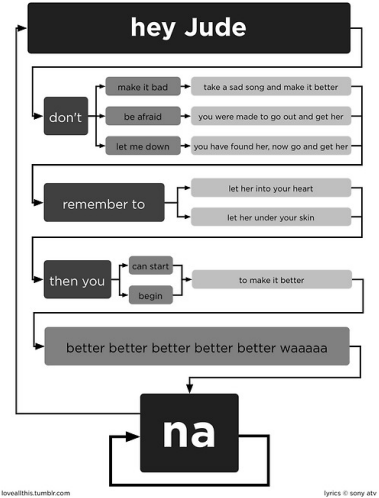
PLEASE PRINT THIS FLOWCHART OUT AND TAPE IT NEAR YOUR SCREEN. CONGRATULATIONS; YOU'RE NOW THE LOCAL COMPUTER EXPERT!

Source(s): https://xkcd.com/627/

Equivalent Pseudocode

```
 1: repeat
 2:     find a related menu item OR pick one at random you haven't tried
 3:     if found one then
 4:         click it
 5:         if it worked then
 6:             done!
 7:         else if been going for > 30 mins then
 8:             give up!
 9:         end if
10:     else
11:         Google a solution
12:         go to 5
13:     end if
14: until done! OR give up!
```

More Interesting Flowchart



Source(s): `http://laughingsquid.com/hey-jude-flow-chart/`

Looking Towards Next Week

- Commencement of *ALL* tutorials/workshops
- You may see more than one tutorial zoom meeting in the LMS, just use the zoom meeting for the tutorial you have enrolled in
- Make sure you can access Grok via LMS; email us if not: `comp10001s2-lecturers@lists.unimelb.edu.au`

Lecture Summary

- Building blocks of programming