

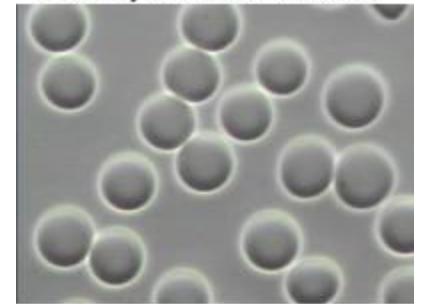
CSc 110

Fundamentals of Programming I

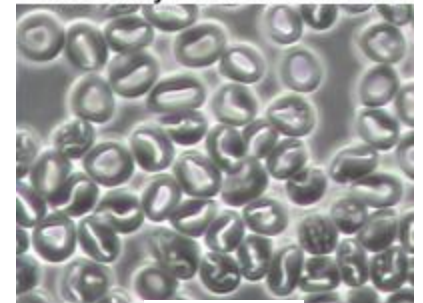
Why Are You Here?

- Did you know that these are healthy blood cells (top) and unhealthy (bottom)?
- An expert only takes a few minutes...
- But multiply those minutes by the number of people who need to be screened...
- There are not enough Drs in the world to check everyone!

Healthy Red Blood Cells



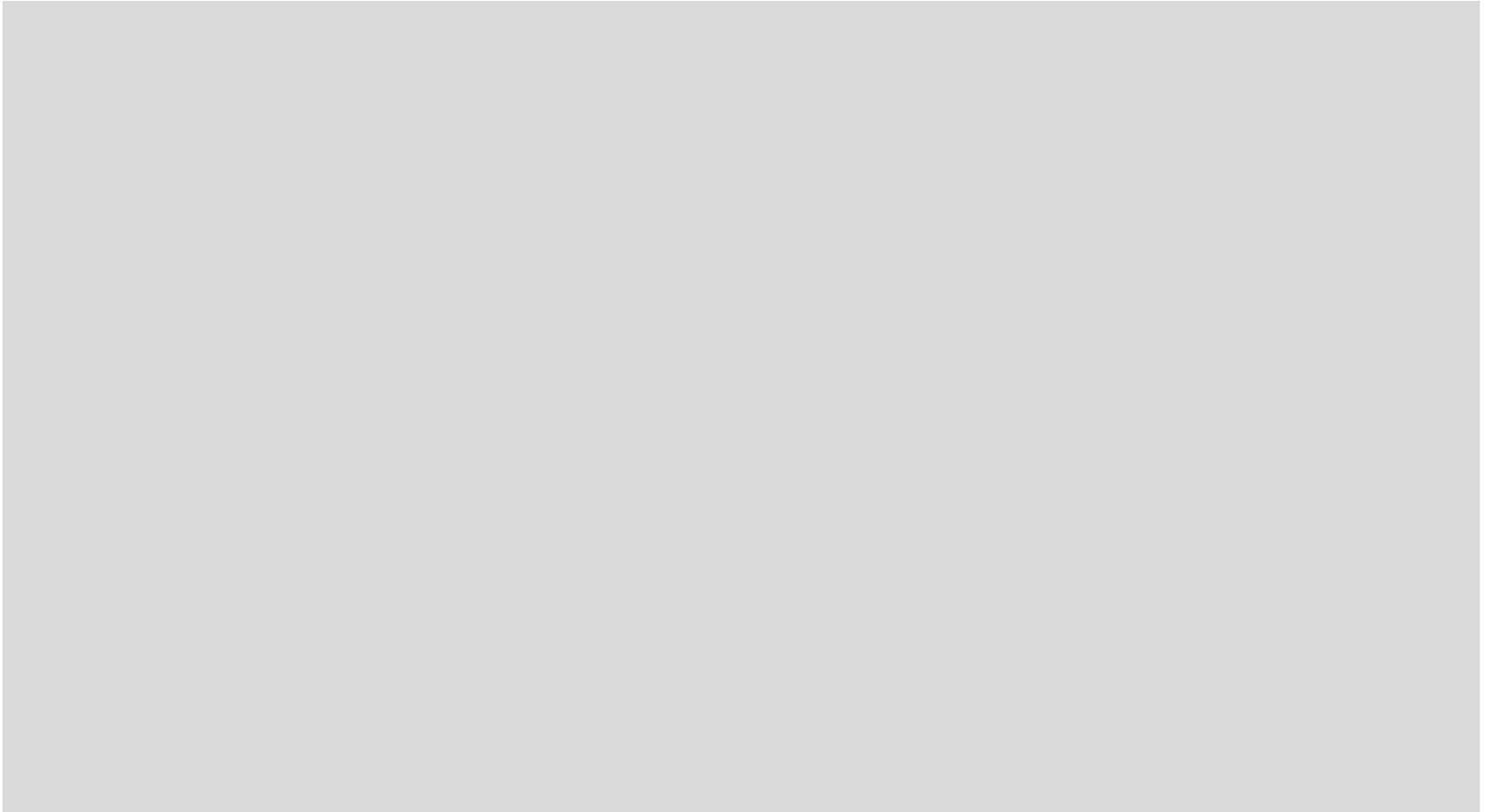
Unhealthy Red Blood Cells



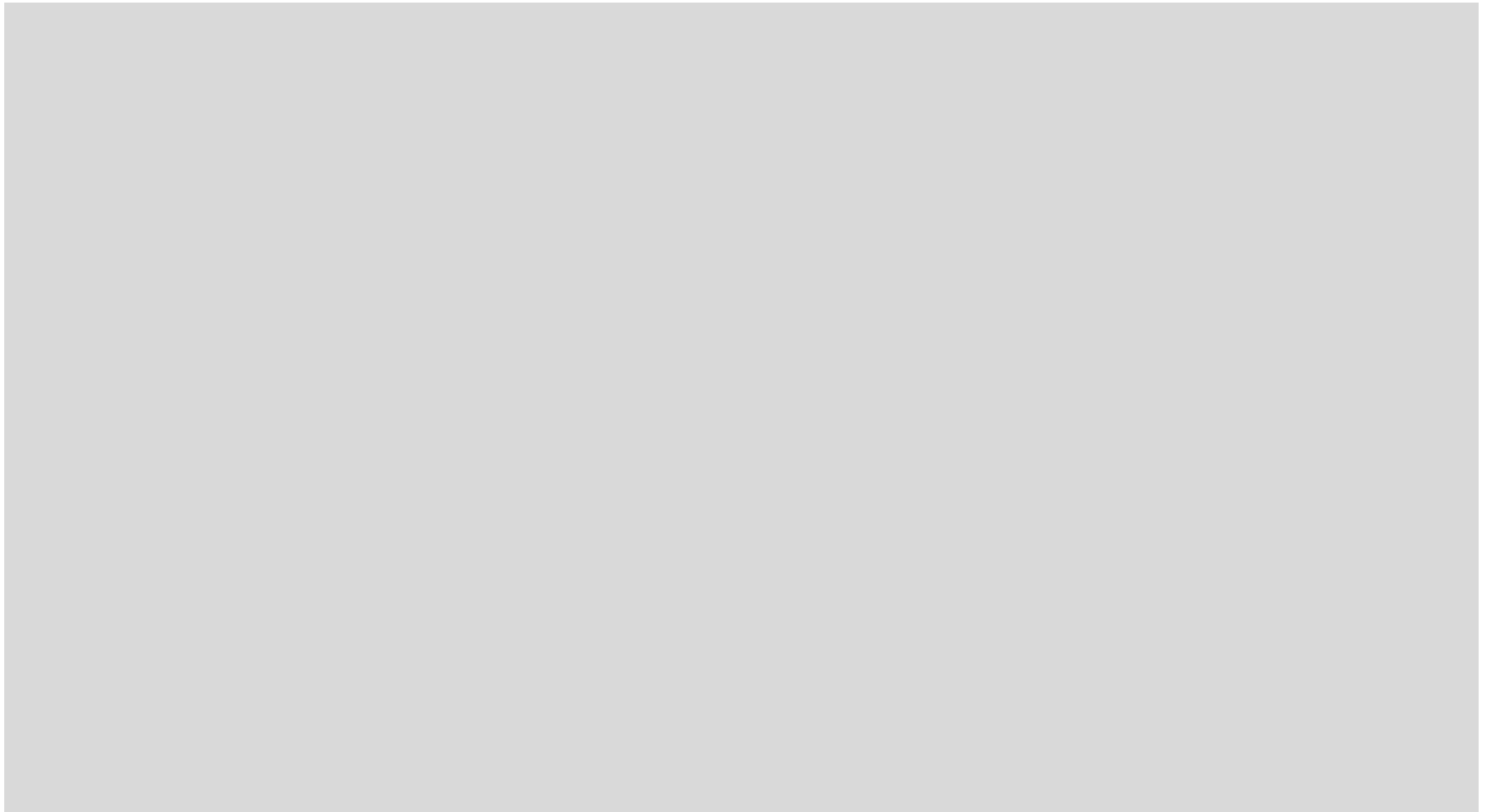
Computer Programming Help?

- Programs can be written to
 - Measure the difference between two pictures
 - Count the number of oddly shaped platelets
 - Analyze gene sequences
 - Analyze the spread of disease
 - Predict the effects of earthquakes
 - Analyze fluctuations in the stock market
 - Track global warming
 - ... connect you to your friends by sending 150 character messages 24 hours a day!

CSc 110: This course is intended to:



At the end of this course,
you will be able to:



CSc 110 Course Structure

- Lectures, Pseudo-Quizzes, Problem Solving Sessions:
 - Ask questions: interactive = student centered.
 - Pseudo-Quiz: Gain experience.
 - Problem Solving Sessions = in class group work.
- 2 hour/week: Hands-on Lab:
 - In the computer lab
 - Its fun!
- Assignments:
 - Lots of 'Self-Check Exercise's and Program Writing
 - Most must be completed independently
 - 2 can be doing by "pair programming"

Csc 110 Overview

- | | |
|---|---|
| <ul style="list-style-type: none">•Computer organization•Syntax & semantics of a high-level language•Variables, expressions, & assignment•Representation of numbers•Input / Output•Conditional & iterative control structures•Functions/methods & parameter passing•Structured decomposition•Problem-solving strategies | <ul style="list-style-type: none">•The role of algorithms in problem-solving•Implementation strategies for algorithms•Debugging strategies•Arrays•Strings and string processing•Data representation in memory•Linear search•Selection sort•Encapsulation & information hiding•Separation of behavior & implementation•Classes |
|---|---|

Plagiarism:

- Submitted work may be checked using plagiarism detection software. Cheating, plagiarism and other forms of academic fraud are taken very seriously by both the University and the Department. You should consult <http://web.uvic.ca/calendar2011/FACS/UnIn/UARe/PolicyAcl.html> for the UVIC policy on academic integrity. Note that the university policy includes the statement that "***A largely or fully plagiarized assignment should result in a grade of F for the course***".

Don't Hand in Other People's Work

Technicalities

Pre-requisite courses:

- If you do not have the pre-requisite course(s) or a waiver for this class then you must drop yourself from the class.
- If they do not, you will be dropped during the first week and a pre-requisite drop is recorded on your record.

Taking the course more than twice:

- If you are taking the class for the third time (or more) you must request, in writing, permission from the Chair of the Department and the Dean of the Faculty to be allowed to stay registered in the class (this is a University Rule).
- This letter should be given to Jane Guy, Undergraduate Advisor in ECS 512.
- If you do not have permission, you will be dropped from the class.

Any questions or concerns? Contact: Jane Guy in ECS 512.

This Course:

- teaches the technical skill of

computer programming

- demonstrates an understanding of and (possibly) changes your

computer culture

Asking Questions & Finding Help

- Your text book
- An alternative book – for your programming language
 - Borrow, beg, bookmark or buy another reference book
- Web Sites
 - Action Item: Find & bookmark some sites
- Asking questions:
 - Classmates (Why not discuss the challenges together?)

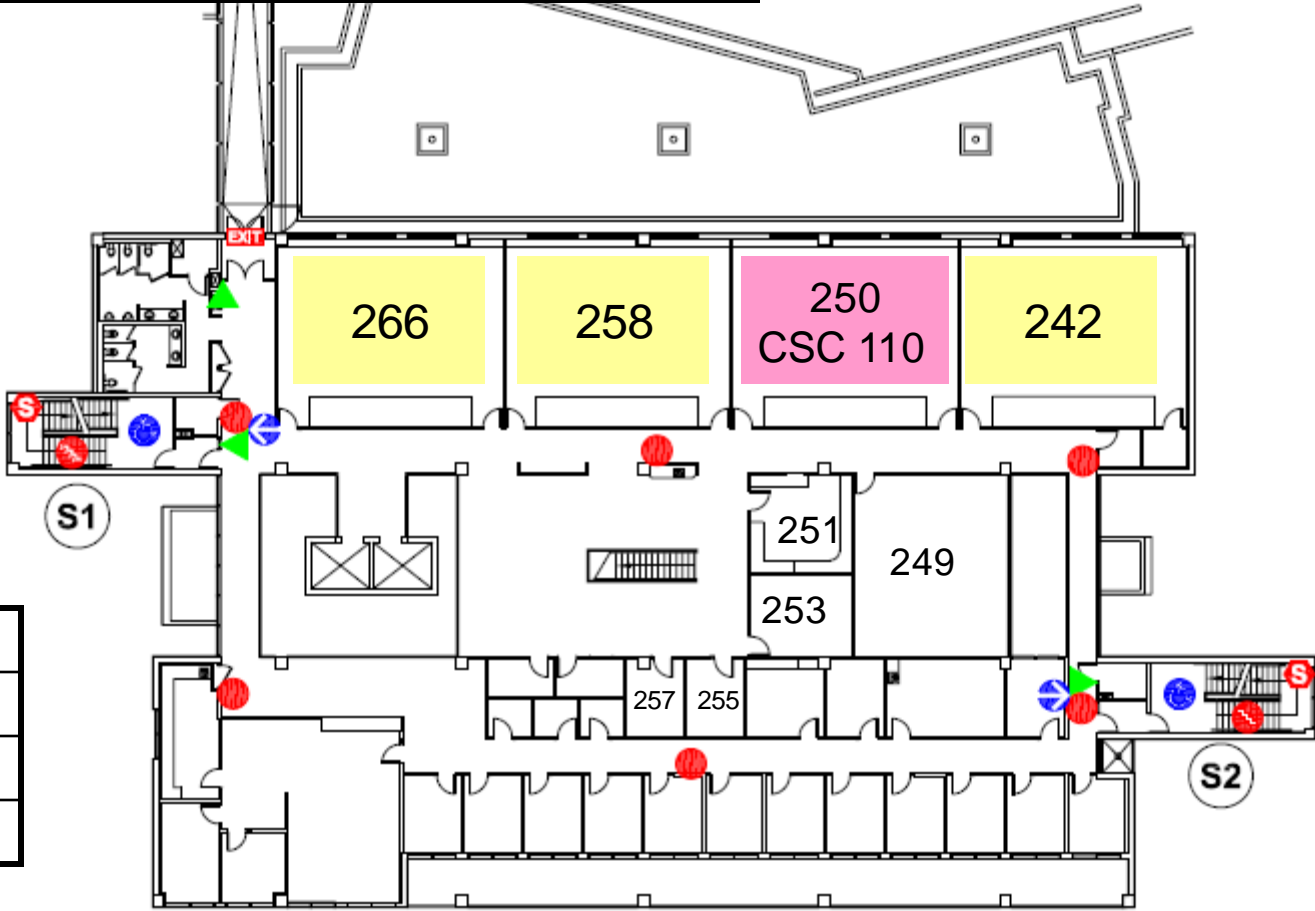
Room 266(30 pcs)	CSC 230 labs
Room 258(30 pcs)	CSC 100, 115 and 160 labs; supp. 225. 330. 340
Room 250(30 pcs)	CSC 105 and 110 labs; supp. 225, 330, 340
Room 242(30 pcs)	CSC 212, 242 and Seng 265 labs (Linux room) ; supp. 330, 340, 360, 370
Room 249-(20 pcs)	Drop-in lab (all windows OS)



250
Your Lab
+ Drop In

Drop In
242, 250,
258, 266,
342, 348, 354

Room 251	Consultant Office
Room 253	TA Office
Room 255	Bill Gorman
Room 257	Victoria Li

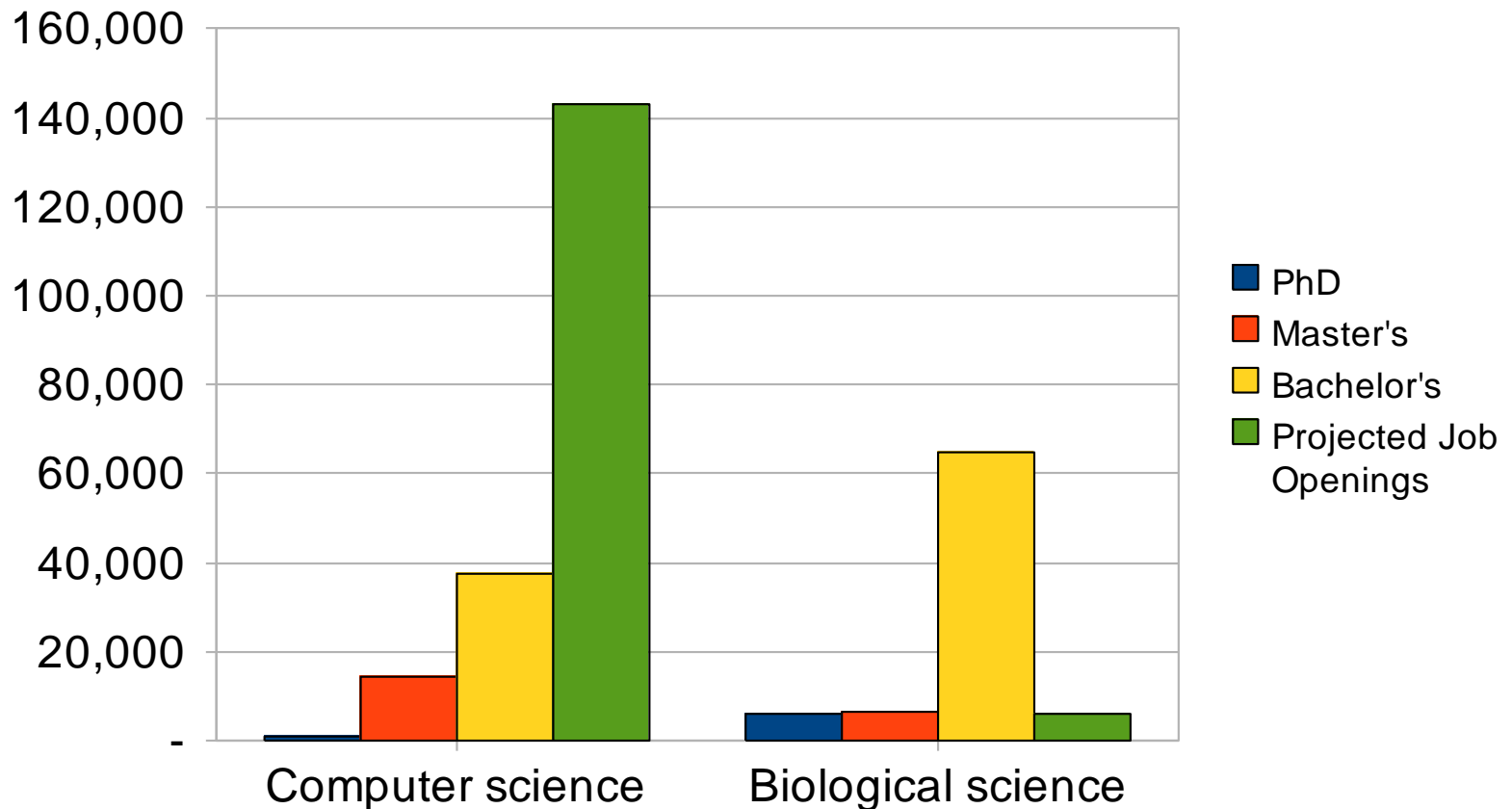


Second Floor

What is computer science?

- Computer Science
 - The study of theoretical foundations of information and computation and their implementation and application in computer systems. -- Wikipedia
 - Many subfields
 - Graphics, Computer Vision
 - Artificial Intelligence
 - Scientific Computing
 - Robotics
 - Databases, Data Mining
 - Computational Linguistics, Natural Language Processing ...
- Computer Engineering
 - Overlap with CS and EE; emphasizes hardware

The CS job market



SOURCES: Tabulated by National Science Foundation/Division of Science Resources Statistics; data from Department of Education/National Center for Education Statistics: Integrated Postsecondary Education Data System Completions Survey; and NSF/SRS: Sur

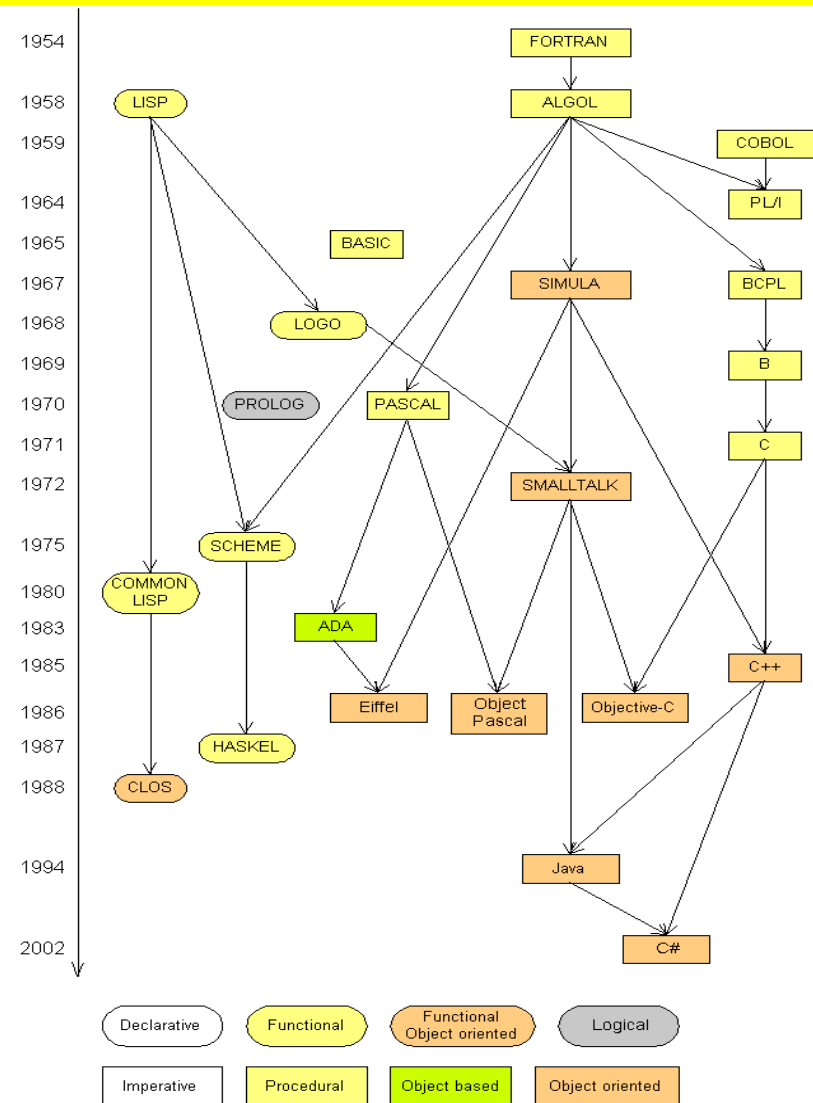
What is programming?

- **program:** A set of instructions to be carried out by a computer.
- **program execution:** The act of carrying out the instructions contained in a program.
- **programming language:** A systematic set of rules used to describe computations in a format that is editable by humans.
 - This textbook teaches programming in a language named Java.



Programming languages

- Some influential ones:
 - FORTRAN
 - science / engineering
 - COBOL
 - business data
 - LISP
 - logic and AI
 - BASIC
 - a simple language



Some modern languages

- *procedural languages*: programs are a series of commands
 - **Pascal** (1970): designed for education
 - **C** (1972): low-level operating systems and device drivers
- *functional programming*: functions map inputs to outputs
 - **Lisp** (1958) / **Scheme** (1975), **ML** (1973), **Haskell** (1990)
- *object-oriented languages*: programs use interacting "objects"
 - **Smalltalk** (1980): first major object-oriented language
 - **C++** (1985): "object-oriented" improvements to C
 - successful in industry; used to build major OSes such as Windows
 - **Java** (1995): designed for embedded systems, web apps/servers
 - Runs on many platforms (Windows, Mac, Linux, cell phones...)
 - The language taught in this course

What is Programming?

- What is a *computer program*?
- What do *computer programmers* do?
- What is *computer science*?
- What do *computer scientists* do?
- What is *software engineering*?

Problem Solving Example: A first “program”

Introduction:

Many encoding programs (encryption, video, sound, etc.) rely on properties of division.

Task:

Write a *predicate* (function that results in “true” or “false”) that decides if an integer d divides evenly into another integer n . For example: `isDivisor(d, n)` is true if ...

Solution Language (#1)

Integer operations: $+$, $-$, \times , \div , `mod`

Relational operators: $=$, \neq , $<$, $>$, \leq , \geq

Integer Constants: 0, 1, 2, 3, ...

Conditional Definition: if

**Group Activity:
Work in groups
of 2 or 3.**

Problem Solving: A second attempt...

Solution Language (#2)

- Integer operations: $+$, $-$, \times , \div
- Relational operators: $=$, \neq , $<$, $>$, \leq , \geq
- Integer Constants: 0 , 1 , 2 , 3 , \dots
- Conditional Definition: `if`

Things to think about:

- How easy was this to solve?
- How confident are you that the solution is correct?

What is programming?

Programming is the act of translating an *idea* for a solution into a clear set of instructions/expressions in a language that can be interpreted by a computer.

To be a *good* programmer you need to know the language well, but you also need to be able to “speak” the language clearly and correctly.

The Joys and Woes of Programming

Joys:

1. sheer joy of making things (even as kids, mud pies!)
2. making things for other people
3. "fascination of fashioning complex puzzle-like objects of interlocking moving parts and watching them work in subtle cycles, playing out the consequences of principles built in from the beginning"
4. joy of always learning
5. delight of a tractable medium "the programmer, like a poet, works only slightly removed from pure thought-stuff. He builds his castles in the air, from air, creating by exertion of the imagination.... Yet the program construct, unlike the poet's words, is real in the sense that it moves and works, producing visible outputs separated from the construct itself... The magic of myth and legend has come true in our time."

Woes:

1. one must perform perfectly
2. other people set the objectives
3. heavy dependence on others (we can make this a joy?!?!)
4. finding little bugs is a lot of work
5. testing drags on and on
6. product is obsolete before completion

"Programming then is fun because it gratifies creative longings built deep within us and delights sensibilities we have in common with all men."

from the mythical man month...