CMPT 125

Introduction

- CMPT 125 and CMPT 127
- Assessment
- Computer science
- Ethics

Administration

CMPT 125 and CMPT 127

CMPT 125

- Theory
 - Algorithms
 - Computer science
 - Analysis
 - Responsibilities of computer scientists
- Assigned work is mostly on paper
- Taught in C 9002
 - Monday 10:30 to 12:20
 - Wednesday 10:30 to 11:20

CMPT 127

- Practice
 - Writing code
 - Debugging
 - Testing
 - Using Linux tools
- Assigned work is on a computer
- Taught in CSIL lab
 - One three hour lab on Tuesday

Assessment

- Assignments 25%
- Midterm 20%
- Final 55% at 8:30am on Saturday 12th August



http://www.cs.sfu.ca/CourseCentral/125/johnwill/

Outcomes

- By the end of this course you can should be able to:
 - Write high quality code in C
 - Use standard command line tools in Linux
 - Use version control to manage your work
 - Develop algorithms to solve problems
 - Predict the behavior of algorithms
 - Reason about the implications of your professional behaviour

Our Expectations

- You are assume to have taken CMPT 120 or have an equivalent background
- We expect you to know about
 - Data types and variable
 - Expressions
 - Basic I/O, from keyboard to monitor
 - Uses of library functions
 - Conditions
 - Repetition
 - Functions and parameter passing
 - Developing, testing and debugging programs

More on Programming Expectations

- You are expected to know how to write programs
 - But not in C
 - In the first few weeks you will learn how to express programming concepts in C syntax



- The expected workload or a three credit course at SFU is ten hours a week
 - CMPT 125
 - Three hours in class
 - Seven hours reading, studying solving assignments
 - CMPT 127
 - Three hours in lab
 - Seven hours coding and experimenting

Brief History of C

- Create in 1972 by Dennis Ritchie of Bell Labs
 - When he and Ken Thompson were designing Unix
 - Developed from Thompson's B language
- Developed to be used as a tool for programmers
 - Working on low level system programs



Computer Science

What is Computer Science?

- The study of algorithms, their formal and mathematical properties, their hardware realizations, their linguistic realizations, and their applications
 - As defined in CMPT 120 or CMPT 130
- TLDR
 - The study of what computers can and cannot do

Then

- One of the earliest (1946) computers was ENIAC
 - Electronic Numerical Integrator and Computer
- It was big
 - 8 x 3 x 100 feet
 - Weighed around 30 tons
 - Required multiple operators
 - Frequently non-functional
- Used to calculate tables
 - Such as artillery firing tables
 - Also used for other complex calculations



Automation

- The industrial revolution led to the automation of tasks previously performed by people
- So too with the advent of the computer
 - Computer and Calculator used to be jobs
 - Performed by people
 - Like the industrial revolution these changes were rapid
 - And are still evolving



and Now



Critical Systems

Computer systems are ubiquitous

- Entertainment
- Communication
- Transportation
- Medicine
- ...
- Many such systems are critical
 - Failure can result in loss of
 - Data
 - Money
 - Life



Responsibilities

- We should take responsibility for our behaviour
- Writing software can have far-reaching consequences
 - So we should behave professionally
- Questions
 - Do you have an obligation to write reliable code?
 - Should you tell your employer if one of your co-workers falsifies tests on his code?
 - What is the right thing to do when your software does not perform correctly and your employer asks you to ignore it?

Ethics

- As adults, we should be able to determine what is appropriate in a given situation
 - This may be difficult to do
 - And may have serious personal or social repercussions
- It is useful to have a framework for reasoning about what is correct, or moral
 - Which leads to the study of ethics