CMPT 125 - D100 Intro.Cmpt.Sci/Programming II

Instructor(s): Igor Shinkar

Calender Objective/Description:
Intro.Cmpt.Sci/Programming II

Instructor's Objectives:
This course is a rigorous introduction to computing science, intended primarily for students who have already taken CMPT 120 as an introduction to algorithms and programming. Students will learn fundamental concepts of computing science, basic principles of algorithm design and software development. All case studies will be presented in C and C++. Students will also learn the basic shell commands in Linux, and will learn how to run and debug their code using VS Code.

Prerequisites:
see go.sfu.ca

Topics:
- Brief review of elementary programming and problem solving; introduction to C.
- Basic shell commands in Linux
- Running/Debugging using Visual Studio Code
- Performance measurements; algorithm design and analysis; asymptotics; fundamental algorithms.
- Encodings of basic types: int, unsigned, float, char, pointer.
- The memory model: addresses, dynamic data types, safe initialization, safe cleanup, and safe arrays.
- Good coding style; defensive coding practices; testing.
- Recursion: simple recursion; recursion on trees; divide and conquer algorithms.
- Reasoning about programs: assertions, invariants, and correctness.
- Abstract data types; information hiding;
- Elementary data structures: stacks, queues, binary trees
- Compound data types; basic object/method design in C++.

Grading:
Final Exam - 35% Lab Exam - 20% Midterm Exam - 20% Homework assignments - 25%

Academic Honesty Statement:
Academic honesty plays a key role in our efforts to maintain a high standard of academic excellence and integrity. Students are advised that ALL acts of intellectual dishonesty will be handled in accordance with the SFU Academic Honesty and Student Conduct Policies (http://www.sfu.ca/policies/gazette/student.html).