CMPT 225

Data Structures and Programming

Course information

- Lecturer: Alireza Khodabakhshi,
  Office: TASC 9211
  Email: alireza@cs.sfu.ca
- TAs:
  - To be announced.

- Course website:
  http://www.cs.sfu.ca/CC/225/ahadjkho/
- Grades:
  https://gradebook.cs.sfu.ca/

Course timetable

- Lectures:
  Mon, Wed, Fri: 3:30pm-4:20pm, AQ 3150
- Midterm: TBA
- Final Exam: TBA
- Office hours:
  - Instructor: (2 hours) Mon, Wed, 1:30pm-2:30pm, TAS9211
  - TA(labs): (3 hours) TBA

Grading scheme

- Midterm: 20%
- Final: 45%
- Assignments: 20%
  (5 assignments each worth 4%)
  Late assignments: -20% each day
- Labs: 12%

  (6 or 7 lab assignments, due on Fridays at 2:30pm)
- AEP (Academic Enhancement Program) Assignment 3%
- Shifting weights:
  - It is possible to shift 50% of the midterm weight (that is 10% of total weight) to final weight.
  - **No other shifts are possible!**

CMPT 225 Topics

- Software Development Process
  - Software Life Cycle
  - Specification, Design and Testing
  - Decomposition, Abstraction and Encapsulation
- Data Structures and Abstract Data Types
  - Arrays and Linked Lists
  - Stacks, Queues and Priority Queues
  - Trees (Binary, Red-Black, Heaps) and Graphs
  - Hash Tables
- Algorithms
  - O() Notation
  - Recursion
  - Sorting Algorithms

Course Objectives

- Develop problem solving techniques
  - Use abstraction to design solutions
  - Design modular programs
  - Use recursion as a problem-solving strategy
- Provide tools for the management of data
  - Identify abstract data types (ADTs)
  - Construct implementations of the ADTs

Problem Solving and Software Engineering

- Coding without a solution design increases debugging time
- A team of programmers is needed for a large software development project
- Teamwork requires:
  - An overall plan
  - Organization
  - Communication
- Software engineering
  - Provides techniques to facilitate the development of computer programs

What is Problem Solving?

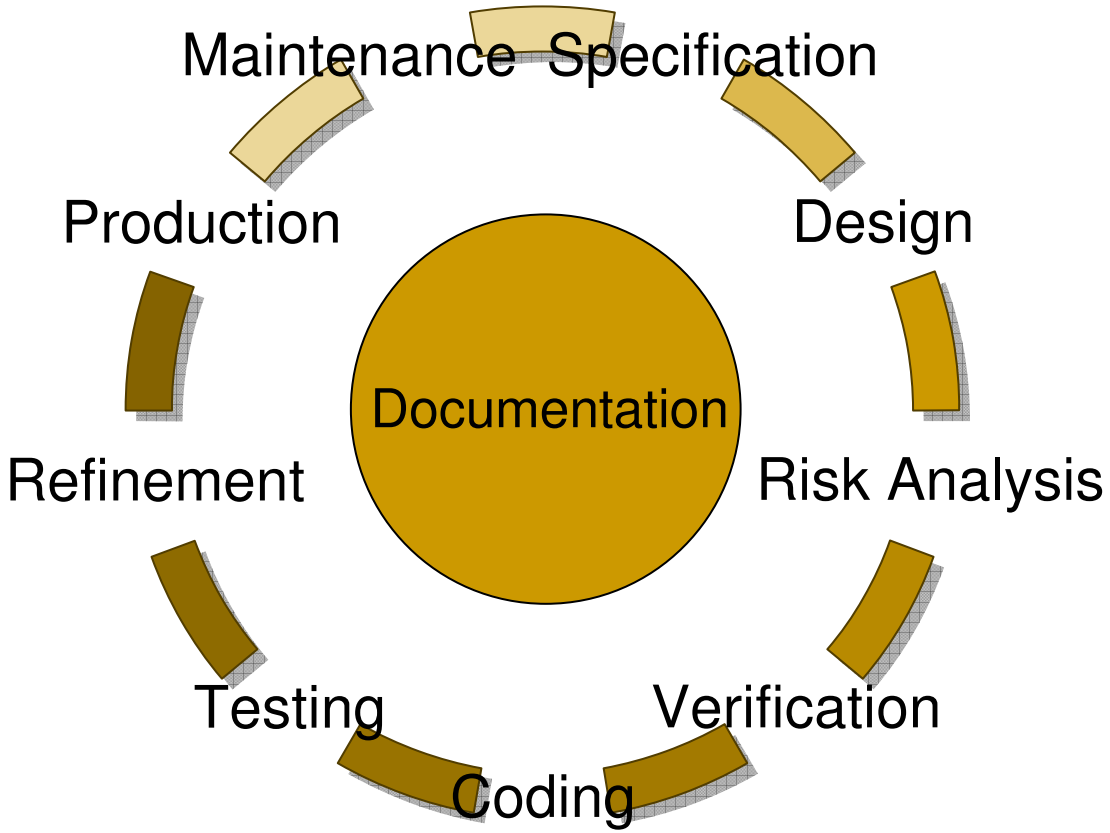
- Problem solving
  - The process of taking the statement of a problem and developing a computer program that solves that problem
- A solution consists of:
  - Algorithms
    - Algorithm: a step-by-step specification of a method to solve a problem within a finite amount of time
  - Data structures to store the data and support the algorithms

What is a Good Solution?

- A good solution is cost effective
  - To develop and maintain:
    - The total cost it incurs over all phases of the development process is minimal
  - To run (i.e. to perform its task)
    - Difficulties encountered by those who use the program (Interaction cost)
    - Computer resources (time and memory) that the program consumes
    - Consequences of a program that does not behave correctly
- Programs must be well structured and documented

Remark: Efficiency is only one aspect of a solution's cost

Software Life Cycle



Software Life Cycle Phases - 1

- Specification
  - Understand the client's problem and requirements
  - Ensure that the requirements are clear and complete and understood by all parties
- Design
  - Plan the implementation of the application's data and operations
  - Plan the testing
- Risk Analysis
- Verification
  - Ensure that algorithms are correct (methods: invariants, etc.)
  - Ensure that the design satisfies the requirements (validation)
- Implementation
  - Write application and test code

Software Life Cycle Phases - 2

- Testing
  - Verify that code works
  - Verify that the code meets the client's requirements
  - There are various types of testing techniques: unit testing, integration testing, system testing, user acceptance testing
- Refining
- Production
  - Package, distribute and install application and train users
- Maintenance
  - Add features
  - Fix bugs
- Documentation
  - Common to *all* the phases of the life cycle
  - Includes the user manual

Software Life Cycle and CMPT 225

- We are primarily concerned with four phases of the life cycle
  - **Design**
  - Verification
  - **Implementation**
  - Testing