CMPT 225
Introducing Abstraction and Abstract Data Type (ADT)
Last Lecture

1. What is this course all about?
2. Course Resources
3. Activity <- Thank you!
4. Expectations
5. Questions?
Learning Outcomes

At the end of this lecture, a student will be able to:

- Define abstraction, information hiding, specifications and "abstract data type" (ADT)
- Differentiate between a class that has been designed as an ADT and a class that has not
- Compare and contrast them
Today’s Menu

- Abstraction
- Information Hiding
- Specifications
- Abstract Data Type (ADT)

Illustrate all these using our Temperature classes
Two Temperature classes

- Have a look at our two Temperature classes on our course web site
- Have a look at each class’ test driver
- Observations?
Abstraction – in the real world

• Abstraction
  ◦ From the Latin abs, meaning away from
  ◦ and trahere, meaning to draw

• Process of taking away or removing characteristics from something in order to reduce it to a set of essential characteristics

Source: http://whatis.techtarget.com/definition/abstraction
Examples of Abstraction

- Car
- Map
  - Where is Vancouver?
Abstraction – in the software world

- Separates purpose of a class from its implementation
  - Abstraction helps identify what should be hidden

- Done during Design
  - Designing a class as an Abstract Data Type (ADT)

Adapted from our textbook: © 2017 Pearson Education, Hoboken, NJ. All rights reserved.
Information Hiding

- We achieve abstraction by hiding information from “public view”
- We hide the implementation
- How?
  - In C++ ->
Specifications

- Describe the purpose of a class and how to use the class
  - Do not indicate how the class is implemented
- We must define a class’ specifications such that client code is able to use the class without knowing its implementation
- How?
  - In C++ ->
Example of Abstraction and ADT

- Temperature class
When a class is an ADT

Client code communicate through slits in the wall
Abstraction

- Why should we hide a class’ implementation?
- Ensure no *client code* can tamper with the class’ hidden details
  - In our Temperature class example, client code can break Temperature class’ invariant
When a class is **not** an ADT

Client code can tamper with a class’ details
ADT – Advantages and disadvantages

- Advantages:
  - 
  - 
  - 

- Disadvantages:
  - 
  - 
  - 
Summary

- Defined abstraction, information hiding, specifications and "abstract data type“ (ADT)
- We can differentiate between a class that has been designed as an ADT and a class that has not
- We know some of the advantages and disadvantages of ADT
Next Lecture

- Data Collection as ADT - Part 1
  - As part of solving a problem (while following the steps of software development process), we shall introduce the data collection List and design it as an ADT