COURSE INTRODUCTION

Programming is a tool for problem solving. In this course, you will learn about new programming techniques that will allow you to solve a wider range of complex problems.

top

COURSE DESCRIPTION

The *University of Manitoba Undergraduate Calendar* describes this course as follows:

An introduction to object orientation, data structures, and algorithms. Prerequisites: COMP 1010 or COMP 1011 (074.101) (C); or High School Computer Science 40S (75%) and any grade 12 or 40S Mathematics.

In the first introductory Computer Science course, you learned how to use a programming language to solve simple problems. This course will build on what you have already learned, showing you a few new features of the language, and several new strategies for applying them. They will allow you to decompose a complex problem into more manageable pieces, to store large "real-world" sets of data, and to manipulate that data with some common but powerful algorithms.

top

COURSE OBJECTIVES

Upon completion of this course you should be able to:

- apply programming techniques learned in COMP 1010 to newer and larger problems;
- use classes and objects effectively;
- write software that deals with large sets of data using files stored on disk;
- create and use collections of data, with an emphasis on the use of classes;
- solve problems using related classes (inheritance);
- apply common algorithms, such as searching and sorting, to collections;
- formulate recursive solutions to problems, and write recursive (self-referential) methods;
- use data structures other than arrays (such as 2D arrays and linked lists);
- recognize when alternate data structures are appropriate to certain kinds of problems; and
- compare the efficiencies of different algorithms used to solve similar problems.

top

COURSE MATERIALS

Required

Bookstore

The following required materials are available for purchase from the University of Manitoba Bookstore. Please order your materials immediately, if you have not already done so. See your Distance and Online Education Student Handbook for instructions on how to order your materials.

- A set of course notes designed specifically for COMP 1020 is available for purchase from the bookstore: Scuse, David. *Growing Algorithms and Data Structures – Third Edition*. 2008.

Required hardware and software

You may use any computer that is capable of running a recent version of the Java Development Kit (JDK) (free download), which includes most recent Windows, Mac, and Linux systems. You should download the *Java Platform Standard Edition (JDK)* version 6 or newer. Note that Mac OS X already comes with the JDK installed and no download is available.

The other requirement is a text editor or other IDE (Integrated Development Environment) for Java. The recommended text editor for Windows is TextPad (shareware), and for other platforms, you can use DrJava (free download). You may use other editors or development tools, but no support will be provided.

We strongly recommend that you install and test all required software immediately. If you encounter difficulties installing your software contact de_support@umanitoba.ca.

Recommended

A good Java reference book is recommended. Any of the textbooks from COMP 1010 (such as Gaddis *Starting Out with Java*) would satisfy this recommendation.

For help with topics from COMP 1010 (such as types, control structures, static methods, and arrays), visit the COMP 1010 wiki.

top

COURSE OVERVIEW

Overview

This course examines the construction of algorithms and data structures using the Java programming language.

Topics

Unit 0 – Review of COMP 1010 concepts:
a refresher on some key topics from the previous course.

Unit 1 – Classes and objects:
combining data and operations in a single unit.

Unit 2 – File input and output:
retrieving data from disk and storing it back again.

Unit 3 – Strings:
more methods and techniques for processing strings.

Unit 4 – ArrayLists and collection classes:
a new way to store collections of data.

Unit 5 – Multi-dimensional arrays:
representing data in two or more dimensions.

Unit 6 – Object hierarchies (inheritance):
improving code reuse by exploiting relationships between data types.

Unit 7 – Sorting:
common algorithms for ordering data.

Unit 8 – Recursion:
using self-referential definition as a problem solving technique.

Unit 9 – Linked lists:
a flexible data structure based on connected elements.

Unit 10 – Abstract Data Types and analysis of algorithms:
abstracting components and comparing techniques.

top

LEARNING ACTIVITIES

Each of the topics in the course will be discussed in week-long unit. The activities for each unit will include readings from the course notes and the online content. At the end of most units, there is a sample problem that will be solved in a step-by-step interactive approach. The final step in each sample problem is an exercise that you will complete and submit early the following week. Alternate units also end with an assignment that must be completed by the posted due dates.

You are also expected to regularly consult the course discussion board. There, you can discuss the course with other students and the instructor, and ask and answer questions about your work (note that there will be restrictions on posting solutions or parts of solutions). The instructor may also use the discussion board to post important announcements or other information.

top

EVALUATION AND GRADING

Your final mark will be calculated according to the following formula.

Item	Percentage
Exercises (Ten in total, each worth 1%)	10%
Assignments (Five in total, each worth 4%)	20%
Mid-term test	15%
Final exam	55%
Total	100%

Note: All final grades are subject to departmental review.

Academic honesty

In this course, you are required to do all of the work on all of the assignments independently. All work that you submit for evaluation must be entirely your own. At no point should you share your work, in whole or in part, with any other students in the class. Violations of this rule are treated very seriously, and have significant consequences. Also note that in the case of shared work, both the person who originally did the work and the person who copied that work are subject to the same penalties.

The single exception to this rule is the code provided for you on the course web site, and in the textbook. You can use the programs you are given, including classes and methods, in whole or in part, in your own work. If you do so, you should appropriately credit the source (indicate in comments where the code originated).

A single *blanket honesty declaration* must be submitted before your first assignment. The statements on that declaration apply to all of your assignments. See the *Assignments* page for more details.

The official description of academic dishonesty follows:

- Academic dishonesty is a very serious offence and will be dealt with in accordance with the University's discipline bylaw. Examples of academic dishonesty include:
  - submitting assignments which are not entirely your own work or providing your work to another student or students
  - use of unauthorized material during a test or examination
  - writing an examination for another person
  - having another person write an examination for you
- Please see the General Academic Regulations and Requirements in the U of M General Calendar for more information.

Note: These policies are also located in your *Distance and Online Education Student Handbook* or you may refer to Student Affairs at http://www.umanitoba.ca/student.

Exercises

Exercises are brief programming activities that occur at the end of each unit in the course. To complete them, you will modify a program that was given to you as part of a larger example.

You do not need to follow the programming standards, though your code should be readable. One mark is awarded for a successfully completed exercise; there are no half marks. If your work is not complete but you submit your solution at least **one week before** the scheduled exercise due date, the instructor will provide feedback that may help you complete the exercise.

Assignments

Each assignment consists of one or more programming questions. Your answers to these questions will be submitted as Java programs.

Assignments are divided into two parts, and include the material from two units. Part A may be started after the completing the first of the two units, and part B will require the material from the second unit. The programs you write in part A may continue in part B, or the may be separate. You will submit your answers to both parts A and B together as your assignment solution.

Assignment 1 covers the units on COMP 1010 review and classes and objects.
Assignment 2 covers the units on file input and output and strings.
Assignment 3 covers the units on ArrayLists and multi-dimensional arrays.
Assignment 4 covers the units on inheritance and sorting.
Assignment 5 covers the units on recursion and linked lists.

Note: Detailed instructions about the assignments are found in the assignment section in your course website.

Assignment and exercise due dates

Consult your course schedule for the assignment and exercise due dates. Late assignments will generally not be accepted.

Midterm test

The midterm test will be conducted in the form of a "take-home" exam, where the test will be distributed on this site and the solutions submitted on a later date. The date of distribution and the due date of the test will be announced by the instructor early in the term.

The test will consist of a variety of short-answer and programming questions, similar in content and length to a timed midterm test written on paper. Your answers will be submitted electronically to the instructor.

SAMPLE FINAL EXAM

There is a sample final exam located with your course materials, or you can access it by clicking on this link "Sample Final Examination".

top

Examination

The final exam will be written at the University of Manitoba (UM), Fort Garry campus or at an approved off-campus location. **Students needing to write at an off-campus location must declare a location by the specified deadline date** (see off-campus declaration and policy under Student Resources on course homepage). **Students writing at the UM Fort Garry campus do not need to declare an exam location.**

The Registrar's Office is responsible for the final exam schedule which is available approximately one month after the start of the course.

The final exam is invigilated, closed-book, and the answers will be written on paper. No computers, calculators, or other aids will be permitted. It will be two hours long and will cover the material from the entire course. It will include short-answer and

programming questions, similar to the type seen on the midterm test. A sample final exam will be provided by the instructor near the end of the term.

Plagiarism, cheating, and examination impersonation

You should acquaint yourself with the University's policy on plagiarism, cheating, and examination impersonation as detailed in the General Academic Regulations and Policy section of the University of Manitoba Undergraduate Calendar. Note: These policies are also located in your *Distance and Online Education Student Handbook* or you may refer to Student Affairs at http://www.umanitoba.ca/student.

Distance and Online Education (DE) Student Resources

In your course website there are links for the following:

- Contact Distance and Online Education Staff
- Distance and Online Student Handbook
- Distance and Online Education Website

Acknowledgments

Content specialist:	John Braico, B.Sc.,MA Computer Science University of Manitoba
Instructional designer:	Sophia Palahicky Distance and Online Education Extended Education Division University of Manitoba
Web publisher:	Lorna Allard Distance and Online Education Extended Education Division University of Manitoba

top