

RYERSON UNIVERSITY

Ted Rogers School of Information Technology Management and G. Raymond Chang School of Continuing Education

COURSE OF STUDY 2017-2018

(C)ITM 200 – Fundamentals of Programming

1.0 PREREQUISITE

The prerequisite for this course is ITM207. Students who do not have the prerequisite will be dropped from the course.

2.0 INSTRUCTOR INFORMATION

- Name:
- Office Phone Number:
- E-mail address:
- Faculty/course web site(s): <https://my.ryerson.ca>
- Office Location & Consultation hours:
 - Your instructor is available for personal consultation during scheduled consultation hours which are posted on their office door or on the course shell in D2L Brightspace. However, you are advised to make an appointment by e-mail or by telephone before coming to ensure that the professor is not unavoidably absent.
- E-mail Usage & Limits:

Students are expected to monitor and retrieve messages and information issued to them by the University via Ryerson online systems on a frequent and consistent basis. ***Ryerson requires that any official or formal electronic communications from students be sent from their official Ryerson E-mail account.*** As such emails from other addresses may not be responded to.

3.0 CALENDAR COURSE DESCRIPTION

This course covers the fundamental principles of object-oriented, event-driven program design and implementation in a business environment. Emphasis will be placed on logic development, program design, modularity, structured programming standards, maintainability, testing and debugging. Specifically, the course will include the following programming features: memory variables; object methods and properties; the logic constructs - sequence, branch, case and loops; simple arrays; basic

file structures; validation and error handling. The course will be taught in a lecture and lab design where a GUI programming language will be used to reinforce the theoretical concepts taught in class.

4.0 COURSE OVERVIEW

The course introduces the fundamental concepts underlying modern computer programming. A systematic approach is used to teach students how to write programs that solve well specified problems. Emphasis is placed on the mastery of basic programming skills, with a considerable attention to the fundamental building blocks of computer programs, and the associated concepts and principles. The essentials of sequential processing and control flow are taught in a procedural programming context prior to introducing classes, objects and related object-oriented programming concepts. To ensure the development of the necessary competencies, assigned homework includes the development of program solutions to problems of adequate complexity and relevance.

5.0 COURSE OBJECTIVES

The learning objectives are:

1. Developing comprehensive knowledge about the fundamental principles, concepts and constructs of modern computer programming.
2. Developing competencies for the design, coding and debugging of computer programs.

6.0 EVALUATION

The grade for this course is composed of the mark received for each of the following components:

Evaluation Component	Percentage of the Final Grade
Weekly Homework	10%
Midterm Exam	30%
Final Exam	60%
Total	100%

NOTE: Students must achieve a course grade of at least 50% to pass this course.

Citation Format for Essays and Term Papers

All essay assignments, term paper and other written works must adhere with APA citation format. Technical errors (spelling, punctuation, proofing, grammar, format, and citations) and/or inappropriate levels of language or composition will result in marks being deducted. You are encouraged to obtain assistance from the Writing Centre (www.ryerson.ca/writingcentre) for help with your written communications as needed.

You can find APA guidelines and academic referencing from the following online resources:

a) Ryerson Writing Support Web site:

<http://www.ryerson.ca/content/dam/studentlearningsupport/resources/citation-conventions/APA%20Basic%20Style%20Guide.pdf>

b) Ryerson Library for APA style guide: <https://library.ryerson.ca/guides/style/>

7.0 POSTING OF GRADES

- ❖ All grades, on assignments or tests must be posted or made available to students through the return of their work. Grades on final exams must be posted. However, as there may be other consideration in the determination of final grades, students will receive their official final grade in the course only from the Registrar. Final official course grades may not be posted or disclosed anywhere by an instructor.
- ❖ Posting of grades on the Course Management System (D2L Brightspace) is preferred. If grades are posted in hard copy they must be posted numerically sorted by student identification number after at least the **first four digits** have been removed. Instructors must inform students in all course management documentation of the method to be used in the posting of grades. Students who wish not to have their grades posted must inform the instructor in writing.
- ❖ Some graded work will be returned to students prior to the last date to drop a course without academic penalty.

8.0 TOPICS – SEQUENCE & SCHEDULE

Session	Topics & Learning Outcomes	Readings	Assignments
1	<p>Introduction to Programming</p> <p>Topics:</p> <ol style="list-style-type: none"> 1. Solving Problems Using Computers 2. Source Code, Compilation and Executable 3. Anatomy of a Simple Java Program 4. Basic Java Programming by Examples <p>Learning Objectives</p> <ol style="list-style-type: none"> 1. Discuss problem solving using computers 2. Explain the difference between source code and executable code 3. Write, compile, run and debug simple Java computer programs 	Chapter 1 & 2	
2	<p>Data Types and Calculations</p> <p>Topics:</p> <ol style="list-style-type: none"> 1. Variables and Data Types 2. Arithmetic Operators 3. Output Statements and Sub-programs 4. Calculations by Examples 5. Strings 6. Getting User Input 7. Input Data Parsing 8. Calculations by Examples 	Chapter 3 & 4	Homework 1
3	Learning Outcomes:		Homework 2

	<ol style="list-style-type: none"> 1. Perform calculations using the usual arithmetic operators 2. Output messages and calculation results 3. Get and interpret user input 4. Perform calculations using user input 		
4	<p>Decision</p> <p>Topics:</p> <ol style="list-style-type: none"> 1. Boolean Expressions 2. Simple Decisions (if-else, switch-case) 3. Nested Decisions 4. Decision by Examples <p>Learning Outcomes:</p> <ol style="list-style-type: none"> 1. Write and evaluate Boolean expressions 2. Use <i>'if-else'</i> and switch-case statements to implement decisions 	Chapter 5 & 7	Homework 3
5	<p>Repetition</p> <p>Topics:</p> <ol style="list-style-type: none"> 1. While Loop 2. For Loop 3. Loop Patterns 4. Combining Control Structures <p>Learning Outcomes:</p> <ol style="list-style-type: none"> 5. Implement repetition (looping) using the “while” and “for” control structures 6. Write programs using nested loops 7. Write programs with a combination of different control structures 	Chapter 6 & 7	Homework 4
6	Midterm		
7	<p>Structured Programming</p> <p>Topics:</p> <ol style="list-style-type: none"> 1. Methods 2. Argument Passing 3. Local vs. Class Variables <p>Learning Outcomes:</p> <ol style="list-style-type: none"> 1. Implement algorithms as self-contained methods 2. Structure programs around multiple communicating methods 	Chapter 8	Homework 5
8	<p>Arrays & Files</p> <p>Topics:</p> <ol style="list-style-type: none"> 1. Arrays 2. File I/O (input/output) 3. Arrays and Files by Examples <p>Learning Outcomes:</p> <ol style="list-style-type: none"> 1. Write programs that make use of arrays 2. Write programs that read from and write to files 	Chapter 9 &10	Homework 6

9	Objects and Classes Topics: <ol style="list-style-type: none"> Preface to Object Orientation What are Objects and Classes Anatomy of a Class Programming with Objects Exception Handling Method Overloading Learning Outcomes: <ol style="list-style-type: none"> Explain the foundational concepts of object-oriented programming Write programs using multiple objects and classes Write programs that include exception handling 	Chapter 11 The Java Tutorial – Object & Classes Tutorials Point - Java Objects & Classes	Homework 7
10			Homework 8
11			Homework 9
12	Advanced Topics Topics: <ol style="list-style-type: none"> Composition Inheritance Abstract Class vs. Interface Learning Outcomes: <ol style="list-style-type: none"> Write programs using composition and inheritance 	Chapter 12 Tutorials Point : Inheritance , Overriding , Polymorphism , Abstraction , Encapsulation , and Interface .	Homework 10

9.0 TEACHING METHODS

The course will incorporate the following teaching/learning methods. A combination of lecture and non-lecture sessions designated at the instructor's discretion. During non-lecture sessions, there will be problem solving laboratory style exercises designed to reinforce the topics taught.

10.0 TEXTS & OTHER READING MATERIALS

Title: Fundamentals of Programming Using Java

Author: Edward Currie

Publisher: Thomson Learning

ISBN: 978-1844804511

11.0 VARIATIONS WITHIN A COURSE

All sections of a course (Day and CE sections) will follow the same course outline and will use the same course delivery methods, methods of evaluation, and grading schemes. Any deviations will be posted on D2L Brightspace once approved by the course coordinator.

12.0 OTHER COURSE, DEPARTMENTAL, AND UNIVERSITY POLICIES

- For more information regarding course management and departmental policies, please consult the '**Appendix of the Course of Study**' which is posted on the Ted Rogers School of Information Technology Management website,

<http://www.ryerson.ca/content/dam/itm/documents/cos/Appendix.pdf>. This appendix covers the following topics:

- 12..1** Attendance & Class Participation
- 12..2** Email Usage
- 12..3** Request for Academic Consideration
 - 12..3.1** Ryerson Health Certificate
 - 12..3.2** Academic Accommodation for Students with Disabilities
 - 12..3.3** Religious, Aboriginal or Spiritual Observance
 - 12..3.4** Re-grading and Recalculation
- 12..4** Examinations & Tests
 - 12..4.1** Period of Prohibition from Testing
 - 12..4.2** Make-Up of Mid-Term Tests, Assignments and Other Assessments
During the Semester
 - 12..4.3** Make –Up of Final Exams
 - 12..4.4** Missing a Make-Up
- 12..5** Late Assignments
- 12..6** Standard of Written Work
- 12..7** Academic Grading Policy
- 12..8** Academic Integrity
 - 12..8.1** Turnitin.com
- 12..9** Student Rights