TED ROGERS SCHOOL Information Technology Management

Ryerson University | The Chang School of Continuing Education

RYERSON UNIVERSITY

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

(C)ITM 200 – Fundamentals of Programming

COURSE OUTLINE FOR 2019-2020

1.0 PREREQUISITE(S)

(C)ITM 207

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:

In accordance with the policy on Ryerson Student E-mail accounts (Policy 157), ***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. 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.

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 OBJECTIVES AND LEARNING OUTCOMES

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.

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.

5.0 TEXTS & OTHER READING MATERIALS

Title: Python for Everybody
Author: Charles R. Severance
Publisher: CreateSpace Independent Publishing Platform
ISBN: 978-1530051120

6.0 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.

7.0 EVALUATION, ASSESSMENT AND FEEDBACK

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

Evaluation Component	Percentage of the Final Grade
Assignments	20%
Midterm Test	30%
Final Exam	50%
Total	**100%**

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

- ❖ At least **20%** of student's grade based on individual work will be returned to students prior to the last date to drop a course in good academic standing .

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.

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:

Student Learning Support > Online Resources > Writing Support Resources

- APA Basic Style Guide

Ryerson Library Citations and Style Guides

- APA Style

8.0 TOPICS – SEQUENCE & SCHEDULE

Session	Topics & Learning Outcomes	Readings	Assignments
1	**Introduction to Computer Programming** **Topics:** 1. Solving Problems using Computers 2. Algorithms 3. Computer Programming 4. Anatomy of a Python Program 5. Basic Python Programming by Examples **Learning Objectives:**	Chapter 1	

	1. Discuss problem solving using computers 2. Explain what are Algorithms 3. Explain what are flow control structures 4. Explain the difference between source code and executable code 5. Explain the difference between interpreted and compiled programming languages 6. Write, run and debug simple Python computer programs		
2	**Variables and Calculations** **Topics:** 1. Variables and Data Types 2. Arithmetic Operators and Expressions 3. Strings 4. Getting User Input and writing output 5. Calculations by Examples **Learning Outcomes:** 1. Read input and write output 2. Perform calculations on user input	Chapter 2 & 6	Assignment 1
3			
4	**Decision** **Topics:** 1. Boolean Expressions 2. Simple Decisions 3. Chained Decisions 4. Nested Decisions 5. Decision by Examples **Learning Outcomes:** 1. Write and evaluate Boolean expressions 2. Use ***'if-else'*** and ***'if-elif-else'*** statement to implement decisions	Chapter 3	Assignment 2
5	**Repetition** **Topics:** 1. While Loop 2. For Loop 3. Loop Patterns 4. Combining Control Structures	Chapter 5	Assignment 3

	Learning Outcomes: 1. Implement repetition (looping) using the "while" and "for" control structures 2. Write programs using nested loops 3. Write programs with a combination of different flow control structures		
6	Midterm		
7	**Functions** **Topics:** 1. Functions 2. Argument Passing 3. Parameters and Arguments **Learning Outcomes:** 1. Implement algorithms as self-contained functions 2. Structure programs using multiple functions	Chapter 4	Assignment 4
8	**Lists** **Topics**: 1. List Methods and Operations 2. Lists by Examples **Learning Outcomes:** • Write programs that make use of lists and strings for data processing	Chapters 8	Midterm marks available on D2L
9	**Files** **Topics**: 1. File I/O 2. Exception Handling 3. File I/O by Examples **Learning Outcomes:** 1. Write programs that read from and write to text files.	Chapter 7	
10	**Dictionaries & Tuples** **Topics**: 1. Dictionaries 2. Tuples	Chapters 9&10	Assignment 5

	3. Dictionaries and Tuples by Examples **Learning Outcomes:** 4. Write programs that make use of dictionaries and tuples for data analysis.		
11	**Objects and Classes** **Topics:** 1. What are Objects and Classes 2. Anatomy of a Class 3. Programming with Objects 4. Class Attributes and Methods 5. Composition 6. Inheritance **Learning Outcomes:** 1. Explain the foundation concepts underlying object-oriented programming 2. Write programs using multiple objects and classes 3. Write programs using composition and inheritance	Chapter 14	
12	**Review**		

9.0 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.

10.0 OTHER COURSE, DEPARTMENTAL, AND UNIVERSITY POLICIES

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

NOTE: Students must adhere to all relevant university policies found in their online course shell in D2L and /or on the following URL: senate-course-outline-policies.

The appendix covers the following topics:

1. Attendance & Class Participation

2. Email Account

3. Request for Academic Consideration
4. Examinations & Tests
5. Late Assignments
6. Standard of Written Work
7. Academic Grading Policy
8. Academic Integrity
9. Student Rights