

# 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 2020-2021

#### 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 virtual consultation during scheduled consultation hours. Information on the consultation format is provided in the D2L course shell. If you wish to make an appointment, kindly do so via email to ensure the professor is available.
- E-mail Usage & Limits:

Students are expected to monitor and retrieve messages and information sent through D2L and Ryerson email on a frequent and consistent basis. In accordance with the policy on Ryerson student email accounts ([Policy 157](#)), Ryerson requires that any electronic communication by students to Ryerson faculty or staff be sent from their official Ryerson email account. Messages from other accounts may be disregarded.

#### 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**

Learning outcomes describe what students are expected to have learned or achieved; as a result, they usually describe what students will be capable of doing, or what evidence will be provided to substantiate learning.

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

Free eBook: <https://www.py4e.com/book>

#### **6.0 TEACHING METHODS**

In Fall 2020 this course will be taught remotely in virtual classrooms. Instruction will take place at scheduled hours, following the approach outlined in D2L Brightspace. You will not be required to attend the Ryerson University campus to complete this course.

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 being 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%
<b>Total</b>	<b>100%</b>

**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](http://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	Chapter 1	

	<p><b>Topics:</b></p> <ol style="list-style-type: none"> <li>1. Solving Problems using Computers</li> <li>2. Algorithms</li> <li>3. Computer Programming</li> <li>4. Anatomy of a Python Program</li> <li>5. Basic Python Programming by Examples</li> </ol> <p><b>Learning Objectives:</b></p> <ol style="list-style-type: none"> <li>1. Discuss problem solving using computers</li> <li>2. Explain what are Algorithms and flow control structures</li> <li>3. Write, run and debug simple Python computer programs</li> </ol>		
2	<p><b>Variables and Calculations</b></p> <p><b>Topics:</b></p> <ol style="list-style-type: none"> <li>1. Variables and Data Types</li> <li>2. Arithmetic Operators and Expressions</li> <li>3. Strings</li> <li>4. Getting User Input and writing output</li> </ol> <p><b>Learning Outcomes:</b></p> <ol style="list-style-type: none"> <li>1. Write programs that read user input, perform calculations using data variables and arithmetic operators and output messages and calculation results.</li> </ol>	Chapter 2 & 6	Homework #1
3	<p><b>Decision</b></p> <p><b>Topics:</b></p> <ol style="list-style-type: none"> <li>1. Boolean Expressions</li> <li>2. Simple Decisions</li> <li>3. Chained Decisions</li> <li>4. Nested Decisions</li> <li>5. Decision by Examples</li> </ol> <p><b>Learning Outcomes:</b></p> <ol style="list-style-type: none"> <li>1. Write programs that make use of Boolean expressions, <i>'if-else'</i> and <i>'if-elif-else'</i></li> </ol>	Chapter 3	Homework #2

	statements to implement decision flow controls		
4	<p><b>Repetition</b></p> <p><b>Topics:</b></p> <ol style="list-style-type: none"> <li>1. While Loop</li> <li>2. For Loop</li> <li>3. Loop Patterns</li> <li>4. Combining Control Structures</li> </ol> <p><b>Learning Outcomes:</b></p> <ol style="list-style-type: none"> <li>1. Implement repetition (looping) using the “while” and “for” control structures</li> <li>2. Write programs using nested loops</li> <li>3. Write programs that combines different flow control structures</li> </ol>	Chapter 5	Homework #3
5	Midterm		Homework #4
6	<p><b>Functions</b></p> <p><b>Topics:</b></p> <ol style="list-style-type: none"> <li>1. Functions</li> <li>2. Argument Passing</li> <li>3. Parameters and Arguments</li> </ol> <p><b>Learning Outcomes:</b></p> <ol style="list-style-type: none"> <li>1. Implement algorithms as self-contained functions</li> <li>2. Structure programs using multiple functions</li> </ol>	Chapter 4	Homework #5
7	<p><b>Lists</b></p> <p><b>Topics:</b></p> <ol style="list-style-type: none"> <li>1. List Methods and Operations</li> <li>2. Lists by Examples</li> </ol> <p><b>Learning Outcomes:</b></p> <ul style="list-style-type: none"> <li>• Write programs that make use of lists and strings for data processing</li> </ul>	Chapters 8	Homework #6  Midterm marks available on D2L
8	<p><b>Files</b></p> <p><b>Topics:</b></p> <ol style="list-style-type: none"> <li>1. File I/O</li> <li>2. Exception Handling</li> <li>3. File I/O by Examples</li> </ol> <p><b>Learning Outcomes:</b></p> <ol style="list-style-type: none"> <li>1. Write programs that read from and write to text files.</li> </ol>	Chapter 7	Homework #7

<b>9</b>	<b>Dictionaries &amp; Tuples</b> <b>Topics:</b> <ol style="list-style-type: none"> <li>1. Dictionaries</li> <li>2. Tuples</li> <li>3. Dictionaries and Tuples by Examples</li> </ol> <b>Learning Outcomes:</b> <ol style="list-style-type: none"> <li>4. Write programs that make use of dictionaries and tuples for data analysis.</li> </ol>	Chapters 9&10	Homework #8
<b>10</b>	<b>Objects and Classes – Part I</b> <b>Topics:</b> <ol style="list-style-type: none"> <li>1. Objects and Classes</li> <li>2. Programming with Objects</li> </ol> <b>Learning Outcomes:</b> <ol style="list-style-type: none"> <li>1. Write programs that use objects and classes</li> </ol>	Chapter 14	Homework #9
<b>11</b>	<b>Objects and Classes – Part II</b> <b>Topics:</b> <ul style="list-style-type: none"> <li>• Class Attributes and Methods</li> <li>• Composition, Inheritance</li> <li>• Examples of Advanced Object-Oriented Programming</li> </ul> <b>Learning Outcomes:</b> <ol style="list-style-type: none"> <li>2. Write programs that make use of advanced object-oriented programming constructs</li> </ol>	Chapter 14	Homework #10
<b>12</b>	<b>Review</b>		

## 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 [Course Outline Appendix](#) 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:

- Attendance & Class Participation
- Email Account
- Request for Academic Consideration
- Examinations & Tests
- Late Assignments
- Standard of Written Work
- Academic Grading Policy
- Academic Integrity
- Student Rights

### Important Resources Available at Ryerson

- [Academic Accommodation Support](#): Ryerson University acknowledges that students have diverse learning styles and a variety of academic needs. If you have a diagnosed disability that impacts your academic experience, connect with Academic Accommodation Support (AAS). Visit the [AAS website](#) or contact [asadmin@ryerson.ca](mailto:asadmin@ryerson.ca) for more information. Note: All communication with AAS is voluntary and confidential, and will not appear on your transcript.
- [The Library](#) provides research workshops and individual assistance. If the University is open, there is a Research Help desk on the second floor of the library, or go to [Workshops](#).
- [Student Learning Support](#) offers group-based and individual help with writing, math, study skills, and transition support, as well as [resources and checklists to support students as online learners](#).
- You can submit an [Academic Consideration Request](#) when an extenuating circumstance has occurred that has significantly impacted your ability to fulfill an academic requirement.
- [Ryerson COVID-19 Information and Updates for Students](#) summarizes the variety of resources available to students during the pandemic.

- Familiarize yourself with the tools you will need to use for remote learning. The [Continuity of Learning Guide](#) for students includes guides to completing quizzes or exams in D2L or Respondus, using D2L Brightspace, joining online meetings or lectures, and collaborating with the Google Suite.